



HAL
open science

Informatique et présentation de soi : la ligne de commande, une esthétique de l'efficacité ?

Oriane Piquer-Louis

► To cite this version:

Oriane Piquer-Louis. Informatique et présentation de soi : la ligne de commande, une esthétique de l'efficacité?. domain_shs.info.comm. 2012. mem_01892970

HAL Id: mem_01892970

https://memsic.ccsd.cnrs.fr/mem_01892970v1

Submitted on 10 Oct 2018

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

UNIVERSITE DE PARIS IV - SORBONNE



CELSA

Ecole des hautes études en sciences de l'information et de la communication

MASTER 2ème année

Mention : Information et Communication
Spécialité : Médias et Communication

« Informatique et présentation de soi : la ligne de commande, une esthétique de l'efficacité ? »

Préparé sous la direction du Professeur Véronique RICHARD

Nom, Prénom : PIQUER-LOUIS Oriane
Promotion : 2011-2012
Option : Médias Informatisés et Stratégies
de Communication
Soutenu le :
Note du mémoire :
Mention :

Cette œuvre est mise à disposition sous licence Attribution - Pas d'Utilisation Commerciale - Partage dans les Mêmes Conditions 3.0 France. Pour voir une copie de cette licence, visitez <http://creativecommons.org/licenses/by-nc-sa/3.0/fr/> ou écrivez à Creative Commons, 444 Castro Street, Suite 900, Mountain View, California, 94041, USA.



Note : Pour toutes remarques, suggestions de pistes éventuelles, trouvailles de fautes de frappe, l'heureuse diplômée répond avec plaisir à *oriane@hauteresolution.net*.

TABLE DES MATIÈRES

| | |
|---|-----------|
| Table des matières | 2 |
| Remerciements | 5 |
| Introduction | 7 |
| Méthodologie | 10 |
| 1 Des imaginaires ambivalents | 15 |
| 1.1 Vu de l'extérieur : une image ambivalente | 15 |
| 1.2 Un discours d'escorte euphorique | 30 |
| 1.3 Des imaginaires artificiels, mais réactivés par les usages | 39 |
| 2 Un usage entre rituel et braconnage | 47 |
| 2.1 Une logique de recommandation | 47 |
| 2.2 Une logique de mise en scène de soi | 54 |
| 2.3 Un « rituel d'usage » | 58 |
| 3 Des rapports de force et des compromis | 63 |
| 3.1 Discipline du corps : une interface dont le pouvoir s'étend jusqu'à notre posture | 65 |
| 3.2 Responsabilité et compromis | 69 |
| 3.3 Esthétique de l'efficacité | 74 |
| Conclusion | 83 |

Cette œuvre est mise à disposition sous licence Attribution - Pas d'Utilisation Commerciale - Partage dans les Mêmes Conditions 3.0 France. Pour voir une copie de cette licence, visitez <http://creativecommons.org/licenses/by-nc-sa/3.0/fr/> ou écrivez à Creative Commons, 444 Castro Street, Suite 900, Mountain View, California, 94041, USA.



| | |
|---|------------|
| TABLE DES MATIÈRES | 3 |
| Annexes | 85 |
| Glossaire | 86 |
| Extrait d'un manuel d'introduction à la ligne de commande : premier chapitre « Put yourself in command » | 88 |
| Extrait d'une présentation d'initiation à la ligne de commande | 92 |
| Une page de manuel sous GNU/Linux | 95 |
| Une liste d'avantages de la ligne de commande publiée sur un blog | 96 |
| Questionnaire établi pour la première salve d'entretiens | 98 |
| Notes du premier entretien | 99 |
| Notes du second entretien | 101 |
| Notes du troisième entretien | 104 |
| Questionnaire établi pour la deuxième salve d'entretiens | 109 |
| Logs de l'entretien, qui s'est effectué sur IRC | 110 |
| Logs d'une discussion sur IRC à propos de la ligne de commande | 117 |
| Résumé et mots-clés | 133 |
| Résumé | 133 |
| Mots-clés | 133 |





Cette œuvre est mise à disposition sous licence Attribution - Pas d'Utilisation Commerciale - Partage dans les Mêmes Conditions 3.0 France. Pour voir une copie de cette licence, visitez <http://creativecommons.org/licenses/by-nc-sa/3.0/fr/> ou écrivez à Creative Commons, 44 Castro Street, Suite 900, Mountain View, California, 94041, USA.

REMERCIEMENTS

Je voudrais d'abord remercier mes deux rapporteurs, Etienne Candell et Jérôme Denis, dont les conseils et l'exigence m'ont permis de mener ce travail à bien.

Mes proches et mes amis ont aidé une aide précieuse : pour les pistes de recherche qu'ils m'ont donné tout au long de l'étude, pour leur indéfectible soutien, pour les « TON MÉMOIRE !!! » à tout bout de champ, merci.

Mon père a eu la gentillesse et la patience de relire ce texte de bout en bout. Merci à lui, cela a été plus qu'utile pour en faire un mémoire digne de ce nom.

Merci également à toute la promotion 2012 du Master MISC, pour la solidarité, les moments de rush et de galère partagés, les coups de pouce...

Merci enfin à Mogmi, ami et auteur du webcomic *Hmm la BD*, qui m'a laissé utiliser son travail à des fins de recherche.





Cette œuvre est mise à disposition sous licence Attribution - Pas d'Utilisation Commerciale - Partage dans les Mêmes Conditions 3.0 France. Pour voir une copie de cette licence, visitez <http://creativecommons.org/licenses/by-nc-sa/3.0/fr/> ou écrivez à Creative Commons, 44 Castro Street, Suite 900, Mountain View, California, 94041, USA.

INTRODUCTION

Dans le cadre de l'étude ethnologique que nous devons effectuer en Master 1 Magistère au Celsa, j'avais choisi d'étudier Le *Loop*, un hackerspace parisien. Un hackerspace –ou laboratoire ouvert– est un lieu où des personnes très diverses, mais ayant toutes en commun d'une part une passion pour l'informatique, l'électronique...et d'autre part une curiosité viscérale qui est à l'origine du terme « *hacker* » : le *hacker* est celui qui déconstruit une technologie, pour comprendre comment elle fonctionne, et ensuite la réutiliser de manière inattendue. C'est en observant ce lieu et les personnes qui le fréquentaient que l'idée de ce mémoire de Master 2 m'est venue : en effet, tous les ordinateurs, quasiment sans exception, étaient sous GNU/Linux et configurés avec une interface en ligne de commande. J'en ai hâtivement conclu que cette interface était l'apanage d'un milieu plutôt expert en informatique, comme celui des *hackers*. Mais peut-on s'arrêter à un constat aussi précipité ?

On peut en douter. En effet, lorsqu'on y prête attention, l'on s'aperçoit vite que l'utilisation de la ligne de commande dépasse le simple cercle des administrateurs système, développeurs, passionnés d'informatique que je croisais dans ce lieu. Il ne s'agit donc pas seulement d'un simple fait qu'on pourrait qualifier de « culturel », puisque dépendant d'un certain milieu social. Des personnes de tous les milieux sont amenées à l'utiliser, des personnes qui ne sont pas forcément « dans l'informatique ». La question qui survient alors est : qu'est-ce qui fait qu'on utilise une telle interface, à l'ère des fenêtres, et icônes très léchées de Mac OS ? Pourquoi n'utilisent-ils pas une interface qui semblerait être celle qui le plus « à la pointe », l'interface graphique, alors même qu'on est dans un domaine, l'informatique, qui fonctionne sur le modèle de l'obsolescence permanente, où l'on avance à coups de mises à jour, qui chassent



l'ancien pour du « mieux »? Nous sommes en effet dans un champ où la dernière version d'un logiciel est vue comme étant la meilleure –au point que la mémoire des vieux logiciels est très difficile à conserver. Si l'on suit cette logique, l'interface en ligne de commande, historiquement plus ancienne, aurait dû être supplantée par les interfaces graphiques. Mais ce n'est pas le cas. Comme l'on voit bien que les utilisateurs de ligne de commande sont loin d'être un groupe d'« irréductibles gaulois » résistant encore et toujours à l'envahisseur, on sent tout de suite que la motivation à utiliser une telle interface dépasse la simple « résistance au changement ».

Mais alors, où trouver un point solide pour démarrer notre réflexion sur cet objet? Comment aborder l'usage de la ligne de commande sans tomber dans le cliché? La solution est peut être de s'intéresser à ces scènes très récurrentes, racontées par des amis informaticiens : ils étaient en train de faire quelque chose de très simple sur leur ordinateur, de tout à fait bénin : discuter sur IRC¹, lire, écrire un e-mail. . .mais comme leur ordinateur était configuré en ligne de commande, les personnes autour d'eux, en voyant l'écran avec le texte en blanc sur fond noir, en concluaient systématiquement qu'ils étaient en train de faire quelque chose de très complexe, de la programmation par exemple. Alors qu'en fait, ce n'est nullement l'action qui est compliquée, mais l'interface qui est juste inhabituelle : nous touchons quelque chose ici qui paraît plus intéressant pour démarrer cette réflexion. L'interface semble fonctionner comme un signe, lié à un certain nombre de choses dans l'imaginaire collectif. Déconstruire les rouages de cette puissance sémiologique peut être un bon point de départ pour notre réflexion. Nous voyons bien qu'il y a un décalage entre l'interface utilisée et l'action menée : c'est là aussi que le problème devient un problème de communication. En effet, c'est la *manière* de faire ce qu'on fait qui, ici, importe plus que l'action effectivement menée. L'interface « communique » quelque chose sur une certaine utilisation de l'informatique. C'est cela que nous allons nous attacher à déconstruire.

Mais ces interfaces ne naissent pas *ex nihilo*. Elles ont été conçues et réalisées par des êtres humains, qui, nous le savons bien, ne sont pas situés hors du monde : ils ont une pensée, une manière de concevoir leur travail, un avis sur l'ergonomie. . .ils sont eux-mêmes nourris par un imaginaire, qu'ils partagent avec les autres, mais également par un imaginaire qui leur est propre. Nous voyons là que le problème se densifie :

1. « Internet Relay Chat », protocole de communication textuelle sur Internet, conçu en 1988



on n'est pas face à un seul paramètre, un imaginaire qui serait partagé par le plus grand nombre, mais face à *plusieurs* discours. Il y a cet étonnement du voisin qui pense que l'on code quand on discute sur IRC, mais il y a aussi le concepteur d'interfaces, voire de système d'exploitation, qui insuffle une pensée, un imaginaire dans son code, celui-là même que l'on exécute pour utiliser cet ordinateur avant même d'avoir fait quoi que ce soit. Il y a une pensée de ce que doit être une bonne interface, et, forcément, une projection de l'utilisateur final. De son côté, l'utilisateur final voit son utilisation de l'interface encadrée par un certain nombre d'imaginaires : ceux insufflés par le concepteur de l'interface tout comme ceux qui circulent dans la société (*via* les films, la publicité...) qu'il va appliquer à l'interface qu'il a choisie. En somme, il y a deux choses à relever à ce stade de la réflexion : le fait que le choix d'une interface soit à l'intersection entre plusieurs paramètres, eux-mêmes complexes, denses et, précisément le fait que cela relève d'un choix. C'est ici que l'on peut avancer l'hypothèse que ce choix peut participer d'une logique de présentation de soi.

Avec ce mémoire, nous allons donc déployer ces réflexions, disséquer les imaginaires concernant l'interface en ligne de commande, les mettre en question. L'enjeu de cette exploration des imaginaires est d'autant plus important que ces derniers participent, du fait qu'ils s'appliquent à une interface, c'est à dire quelque chose qui doit s'effacer pour nous permettre d'utiliser comme nous le désirons nos machines, de ce que Georges Perec a appelé « l'infra-ordinaire », ce qui fait partie de notre quotidien au point qu'on en a à peine conscience, et qui pourtant, influence la manière dont nous vivons.

La question que nous pouvons nous poser est donc : « *Jusqu'où l'utilisation d'un certain type d'interface peut influencer le rapport à l'informatique ?* ». Voilà la direction que prendra notre travail. Pour préciser encore cette question, nous nous demanderons si, dans le cas de la ligne de commande, cette influence peut prendre la forme de ce que nous appellerons une « esthétique de l'efficacité » (qu'il nous faudra définir). Nous essaierons de voir les logiques de pouvoir à l'oeuvre et la marge de manoeuvre qu'il est possible de dégager par rapport à d'éventuels schémas imposés.

Pour répondre à notre question, il nous faudra aller et venir entre trois pôles de réflexion :



- une réflexion sur les *usages* : jusqu'où les usagers utilisent la ligne de commande telle qu'elle a été faite pour ?
- une réflexion autour du concept d'*image du texte* développé par Emmanuel Souchier, qui semble particulièrement pertinent ici ;
- une réflexion plus politique, inspirée par Foucault, pour penser ces interfaces en ayant à l'esprit leur pouvoir contraignant, et l'étendue de ce pouvoir ;

Ces trois pôles ne constituent pas à part entière un plan, mais la réflexion ira de l'un à l'autre : ils sont beaucoup trop liés pour être pris les uns à part des autres. Il faut les mêler pour mieux saisir la complexité de l'objet que nous étudions.

Il s'agira bien d'observer l'interface en ligne de commande en tant qu'objet communicationnel, c'est à dire pris dans son rôle d'interface :

- entre un utilisateur et sa machine,
- entre un individu et son environnement social (puisque l'informatique n'est pas une activité hors du monde).

Méthodologie

Point de vue

Ce mémoire est le digne héritier de mon étude ethnologique au *Loop*. Lors de cette étude, j'avais adopté une position un peu hybride : déjà très technophile, acquise à plusieurs points de ce qu'on appelle « l'éthique *hacker* », mais étudiante en Sciences de l'Information et de la Communication, et amenée à porter un regard que je me suis efforcée de rendre différent et distancié. Le fait que je connaisse des éléments de la culture *hacker* m'a aidé à comprendre des choses que je n'aurais certainement jamais comprises sinon. Mais la distance était nécessaire pour produire un document à prétention scientifique. J'adopte pour ce mémoire le même genre de position : ma fréquentation du hackerspace a laissé quelques séquelles et j'ai depuis moi-même un ordinateur sous GNU/Linux, avec une configuration majoritairement en ligne de commande. J'ai eu la chance de pouvoir configurer mon poste de travail au bureau exactement de la même manière. De l'autre côté, je suis toujours étudiante en MISC, et je garde toujours à l'esprit mon mémoire et cette rigueur scientifique que l'on nous a enseigné au Celsa : ne jamais rien prendre pour acquis et tout remettre en



question. La manière dont j'ai collecté les informations nécessaires à ma réflexion est fortement influencée par ce positionnement : je me suis efforcée d'apporter à mon activité, pour le bien de ce mémoire, un regard de l'ordre de l'observation participante en ethnologie.

Connaître le point de vue des usagers et de leur entourage

Afin de mieux comprendre le rapport que les usagers entretiennent avec leur machine, j'ai envisagé des entretiens. Je souhaitais toucher un panel assez large, pour avoir d'une part des retours de la part d'usagers de la CLI (*Command-Line Interface*, interface en ligne de commande), mais aussi de GUI (*Graphical User Interface*, interface graphique), pour bien comprendre les raisons du choix de l'un et de l'autre et évaluer jusqu'où les logiques de milieu sont responsables de l'adoption de tel ou tel espace de travail. La plupart des entretiens réalisés sont plutôt des entretiens exploratoires, mais ils composent déjà une matière pour ma réflexion. Entretiens exploratoires et directs se combineront ainsi : les premiers auront recentré le terrain de ma recherche et les seconds permettront de vérifier mes hypothèses, qui sont bien fixées maintenant. Le panel estimé était de dix à quinze personnes pour des entretiens qualitatifs, exploratoires ou directs. En réalité, seulement quatre entretiens (trois exploratoires et un semi-directif) ont pu être réalisés, avec une distortion manifeste dans ce panel réduit : tous sont des usagers de la CLI. Leurs discours ont été accueillis et analysés avec la distance qui s'imposait. Parallèlement à ces entretiens, je me suis aperçue au cours de mes recherches que les canaux IRC que je fréquentais étaient une source d'information intéressante. Parfois, au détour d'une conversation, une personne va évoquer son utilisation de la ligne de commande, la réaction de l'entourage, ou des souvenirs d'interfaces antérieures. Ce travail est donc également nourri d'une observation complémentaire sur IRC.

Connaître le point de vue des concepteurs

Cela permet d'évaluer ce qui est insufflé comme idées et comme projection de l'utilisateur dans l'interface. De nombreux documents sont publiés par l'équipe de design de Canonical (l'organisation à l'origine de la distribution Linux grand public



Ubuntu), notamment des chartes et des réflexions sur le design orienté-utilisateur. Ils permettent de se faire une bonne idée de ce qu'un concepteur d'interfaces a comme préoccupations. Ubuntu a été choisi parce que cette distribution Linux a un positionnement assez particulier : c'est une distribution Linux, qui hérite à ce titre d'une certaine vision de l'informatique, marquée par une très grande place donnée à la ligne de commande, mais dont le design est résolument « orienté-utilisateur » avec une interface graphique très léchée, qui se rapproche assez du travail d'Apple. Il me semblait intéressant de connaître leur point de vue du fait de ce positionnement. Les travaux engagés dans les années soixante à Xerox PARC où sont apparues les premières interfaces graphiques, ainsi que la souris (notamment ceux de Douglas Engelbart) sont également une source intéressante d'information.

Des analyses sémiologiques

Il nous a semblé intéressant d'étudier ce qui fait sens dans l'image des différents espaces de travail. Aussi nous avons étudié les copies d'écran de bureaux de personnes utilisant (ou non) la ligne de commande, et ce sous divers systèmes d'exploitation. Cela permet de bien voir les différents imaginaires délivrés par les différents systèmes d'exploitation et comment ces imaginaires rentrent en relation avec la présence ou non d'une interface en ligne de commande, et comment cette interface est pratiquée. Nous avons aussi analysé certains programmes qui utilisent un « look » voulu comme étant très proche de celui d'un terminal pour voir comment cela est mis en scène.

Interroger les imaginaires

Faire l'étude d'extraits de films ou de séries présentent des images de terminal peut également être intéressant. En effet, le cinéma charrie un certain nombre d'imaginaires qui sont ensuite intégrés, même par les usagers de la CLI. Cela a consisté en l'analyse de scènes clés (pour comprendre comment sont présentés les personnages, dans quel type de contexte intervient la ligne de commande) et l'analyse sémiologique de captures d'écran.

Parmi les films choisis pour l'étude d'extraits et d'images, nous avons :



- *Sneakers*, de Phil Alden Robinson : un film des années 90, qui raconte l’histoire de deux jeunes génies de l’informatique. Ici, c’est clairement la place de la ligne de commande dans l’imaginaire du *hacker* au cinéma que l’on a voulu observer.
- *The Matrix*, des frères Wachowski : ce film des années 2000 semble particulièrement intéressant, car les personnages du films sont en quelque sorte enfermés au sein d’un programme informatique, la *Matrice*, qui est très présent dans le film. Il s’agissait de voir ici comment la ligne de commande est mise en scène.
- *Tron*, réalisé par Steven Lisberger en 1982, et *Tron Legacy*, réalisé par Joseph Kosinski en 2010 : l’analyse de ces films va dans le même sens que celle de *Matrix*, à la différence qu’on dispose en regard d’un corpus écrit : un des concepteurs graphiques de *Tron Legacy* explique sur son blog son travail sur la ligne de commande pour le film. L’autre point intéressant à observer ici était l’évolution du film entre l’original et le « remake », à quarante ans d’écart.
- *Millénium* : nous en avons visionné des extraits, pour observer comment la ligne de commande est insérée dans le contexte et la dramaturgie du film, ainsi que dans la construction du personnage de Lisbeth, l’héroïne.

Nous avons volontairement choisi des films à des époques différentes, pour observer s’il y a une progression. Cela donne également une cohérence chronologique au corpus.

Observer

Dernier point de la méthodologie : l’observation. Comme cela est difficile à mettre en place dans le temps imparti, nos observations sont surtout basées sur les notes prises lors de l’étude ethnologique réalisée sur le *Loop* en 2011, qu’on a complétées par d’autres observations et des recueils de témoignages pour mieux comprendre les usages, à la fois dans le rapport usager-machine, et dans le rapport usager-machine-société, pour comprendre les logiques de présentation de soi.

La lecture de Bruno Latour m’a donné des points de méthodologie qui ont complété ceux glanés lors de la lecture de l’article d’Emmanuel Souchier sur l’image du texte. Son approche de la sociologie permet plusieurs choses, appliquées à un mémoire professionnel de M2 écrit en période de stage : d’une part, il prend pleinement en compte les mouvements et le comportement de l’étudiant sur le terrain, et d’autre



part, il donne un guide intelligent pour « suivre » des usages –ce qui est précisément le but de ce mémoire : étudier les *traces* laissées par les usages et les usagers. Enfin, la méthode d'étude suggérée par ANT met en lumière que les objets ne sont pas neutres, et participent au « social », ce qui est également tout à fait le point de cette étude.

J'ai donc opté pour une méthodologie assez souple, qui consistait en la collection de traces, grâce auxquelles j'ai essayé de comprendre et d'établir les liens entre tous les « mouvements » observés. Ces traces sont : des captures d'écran, des images de films, des verbatims recueillis en entretiens, des notes prises sur le terrain (celles que j'ai prises lors de mon étude ethnologique l'année dernière et celles que j'ai pu glaner au Nicelab, le hackerspace de Nice où je me rendais régulièrement).

Selon la même méthodologie, l'on s'aperçoit vite que travailler sur la ligne de commande rend inévitable le traitement de réseaux de concepts connexes, avec d'une part tout ce qui concerne les interfaces graphiques et d'autre part ce qui concerne Linux et les logiciels libre. Il faut également se pencher sur l'image du *hacker*, du geek. On est forcé de rechercher dans un spectre plus large pour bien cerner le problème sans en omettre des aspects déterminants.

Concernant la méthodologie, il m'a semblé également important de mettre en lumière la façon dont j'ai procédé en écrivant ce mémoire. J'ai opté pour une écriture en « spirale », où j'écrivais souvent –j'ai commencé assez tôt à rédiger le mémoire en lui-même– et non pas de manière linéaire, mais circulaire, afin de toujours vérifier si mon raisonnement tenait la route, et afin de bien diriger ma propre réflexion. L'écriture, en effet, stimule et fixe la pensée. Il me semblait intéressant de fonctionner de cette manière pour fournir un travail de qualité. Travailler de cette manière oblige également à se relire souvent (et non pas seulement à la fin de la rédaction du mémoire) et de vérifier si la qualité littéraire du rendu ne nuit pas à la compréhension de la réflexion.



1

Des imaginaires ambivalents

« *La création de machines et dispositifs est toujours encadrée par un imaginaire.* »¹

C'est avec cette citation extraite de la thèse de Bernhard Rieder que nous démarrons notre réflexion. Nous avons en effet décidé de nous concentrer d'abord sur les imaginaires qui entourent notre objet, pour évaluer leur pouvoir d'influence sur les usagers. Le constat de Rieder est on ne peut plus juste, surtout en matière de technologie : l'imaginaire, l'idée qu'on se fait d'un outil ou d'une interface, les discours en général euphoriques sur cet outil ou cette interface, sont des moteurs de l'innovation. Si nous n'avions pas eu toutes ces images populaires sur un futur où il y aurait de la visiophonie, nous n'aurions peut-être pas développé aussi bien les technologies attendant à la vidéo en temps réel. Ce sont des imaginaires qui ont influencé le développement de cet outil. Si c'est le cas, ce sont également des imaginaires qui ont influencé l'adoption de la ligne de commande. Voyons de quoi ces imaginaires sont faits et quelle est leur pouvoir d'influence.

1.1 Vu de l'extérieur : une image ambivalente

La ligne de commande est donc, en vertu de ce que nous venons d'exposer, un objet sémiotique, avant d'être un objet technique, puisqu'elle est investie de discours, d'images, d'idées (les imaginaires) avant même d'être utilisée. C'est un signe dont la force se remarque au quotidien : quelles que soient les réactions, utiliser la ligne de commande, que ce soit pour lancer une suite de commandes dans un terminal ou

1. RIEDER, Bernhard, *Métatechnologies et délégation*, Paris, Université Paris VIII, 2006, p.47



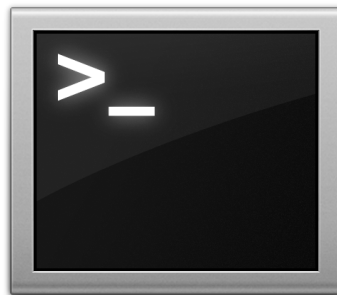
un programme en « mode console » (c'est à dire sans autre interface que celle du terminal), ne laisse jamais indifférent.

Avant d'essayer de comprendre quels imaginaires nourrissent ces réactions, déclinons tout d'abord les caractéristiques de la ligne de commande en tant que signe :

- du texte blanc (ou vert) sur fond noir
- l'absence de menus
- une typographie très simple, proche de celle des anciennes machines à écrire
- le *prompt* (qui affiche le nom d'utilisateur, celui de la machine, et le dossier courant, à gauche) et le curseur qui clignote

Pour bien illustrer notre propos, nous pouvons prendre comme base l'icône « terminal » de la distribution Ubuntu (*figure 1.1*), qui rassemble toutes ces caractéristiques et qui, par son statut d'icône métonymique montre bien que la ligne de commande en tant que signe se résume à cela.

FIGURE 1.1: Icône du programme « Terminal » pour Ubuntu



Le terminal y est réduit à sa plus simple expression : le *prompt* est représenté par le symbole « > » et le curseur par un « _ ».

Nous pouvons remarquer que ce n'est pas le programme ou l'activité à l'œuvre qui sont importants, mais l'apparence que le programme a à l'écran. Les réactions que suscite la ligne de commande sont moins causées par ce que l'on pratique réellement que par une certaine esthétique, qui évoque à elle seule ce qu'on est supposé faire sur son ordinateur en utilisant cette ligne de commande. D'ailleurs, les discours que nous avons recueillis soulignent le fait que celui qui utilisait son ordinateur *via* la ligne de commande était, dans la plupart des cas, en train d'effectuer un activité anodine, comme discuter sur IRC.



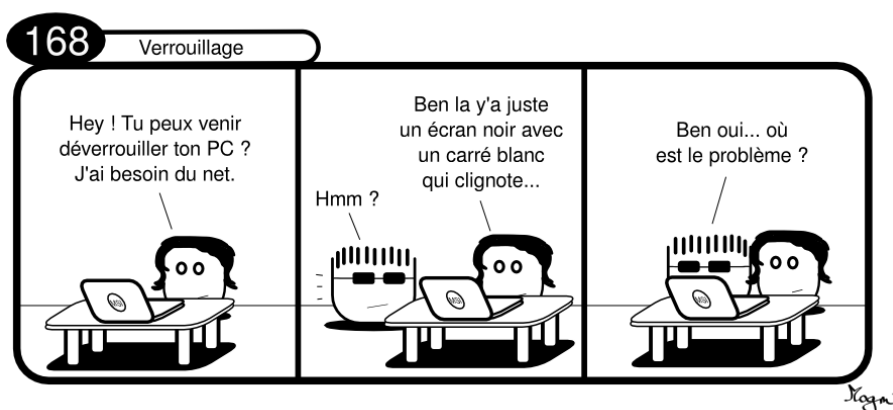
Voici, pour exemple, ce que l'on peut lire sur IRC, lorsqu'on demande quelles sont les réactions les plus fréquentes de leur entourage à des utilisateurs de ligne de commande :

« Avec le péon moyen, ça va du "Han, le pauvre, son PC est tout pété" à "Au secours, il est en train de pirater mon ordi" ou "Tu peux pirater l'ordi du prof, dis?" - alors que je suis juste en train d'utiliser Vim dans une TTY² hein. »

ou encore :

« De vieux jeux vidéo, je comprends, mais pourquoi dépenser autant pour un ordinateur aussi vieux? »

Une situation tellement emblématique de la relation entre l'informaticien et son entourage proche que l'on retrouve la scène dans le webcomic *Hmm La BD* :



Que mettent en lumière ces réactions ?

Il s'agit là d'une première interprétation à chaud de ce qu'on pourrait appeler le signe « ligne de commande » : l'on ne réagit pas par rapport à l'objet, mais à ce qu'il évoque pour nous, l'image qu'on en a. Il est bien question d'imaginaires, d'*a priori*, et c'est cela qui nous intéresse ici.

Nous ne citons que deux réactions parmi tant d'autres : cette anecdote est assez répandue. En posant la même question sur IRC à des gens très différents, les mêmes types de réponses reviennent, du simple *smiley* « :o » (qui signifie l'étonnement) à la liste des réactions rencontrées.

L'on a plusieurs types de réponses récurrentes :

2. Vim est un éditeur de texte en « mode console », et une TTY est un terminal lancé sans aucune interface graphique. Voir Glossaire.

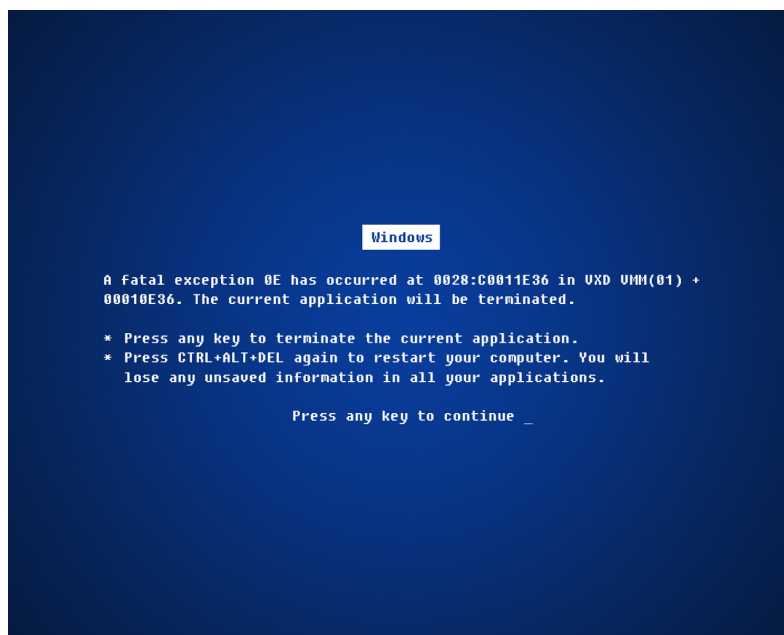


- ton ordinateur est cassé
- ton ordinateur est vieux
- tu es en train de faire quelque chose d'illégal
- tu es en train de faire quelque chose de très compliqué

Regardons maintenant ce que contiennent en termes d'imaginaires chacune de ces réactions-type.

Tout d'abord, l'assertion comme quoi l'ordinateur est cassé : il se trouve que lorsque le système d'un ordinateur sous Windows ne fonctionne plus, ou très mal, il quitte toute interface graphique et affiche un message d'erreur en blanc sur fond bleu. Cette erreur est connue sur Internet sous le nom : « Blue Screen of Death » ou « l'écran bleu de la mort ». L'absence totale d'interface graphique, l'aspect dépouillé de la police et la violence du contraste font que les utilisateurs de Windows (qui reste le système d'exploitation le plus largement utilisé) associent à ce type d'interface l'idée que la machine a un problème.

FIGURE 1.2: « Blue Screen Of Death »



Ensuite, la vétusté de la machine : la ligne de commande est effectivement un outil daté, dont l'apparence n'a pas bougé –ou si peu– c'est donc normal de penser que la machine elle-même est datée. A défaut d'avoir en tête l'image des écrans des ordinateurs des années soixante, nous avons en tête des images venues du cinéma, par exemple, qui nous font associer obsolescence et ligne de commande. Le film



Wargames, par exemple, met en scène des personnes utilisant des ordinateurs sans interface graphique (figure 1.3), puisque Xerox PARC, d'où sont sorties les premières interfaces graphiques, était en pleine ébullition, mais ces interfaces n'étaient pas encore sur le marché. L'ordinateur (figure 1.4³) utilisé dans le film est effectivement un très vieux modèle. Le film étant sorti en 1983, l'on peut avoir de bonnes raisons de penser que l'interface était ainsi parce que l'informatique n'était pas aussi développée qu'aujourd'hui.

FIGURE 1.3: Image extraite du film *Wargames*

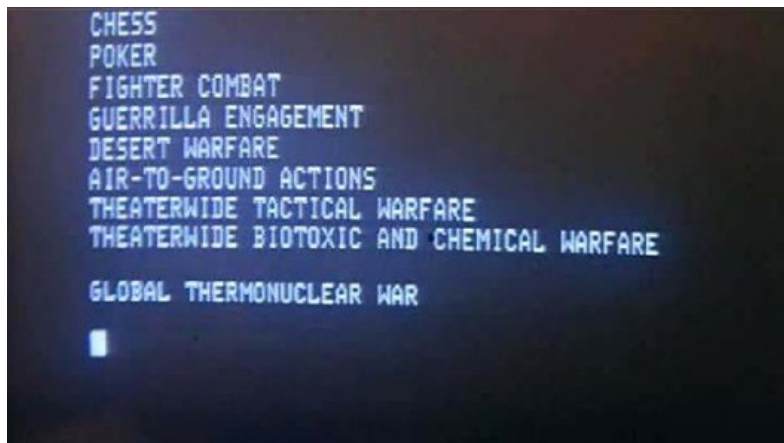


FIGURE 1.4: L'ordinateur utilisé dans *Wargames*, l'Imesai 8080



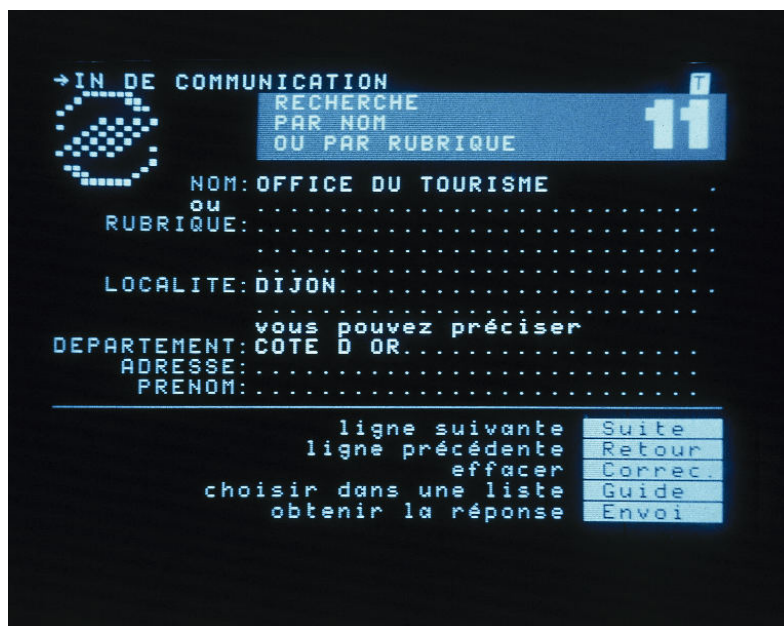
Remarquons que, en plus de ce type d'image, la plupart des Français ont à l'esprit l'interface du Minitel, objet aujourd'hui désuet mais très présent dans l'imaginaire

3. Photo : *Wired* / Todd Hido, disque dur Imesai FDC2-2 fourni par *vintagetech.com*.



populaire, même pour une génération qui l'a peu voire pas connu. L'interface du Minitel est très proche de celle d'un *shell*. Même contraste fond noir / texte blanc, même typographie très dépouillée. De plus, le Minitel n'était pas doté de souris, le curseur ne pouvait se mouvoir que grâce au clavier. Toute personne qui a utilisé un tant soit peu cet outil est tenté de projeter l'expérience quelque peu frustrante de l'utilisation du Minitel sur celle de la ligne de commande. Le Minitel étant très clairement, aujourd'hui, le signe de l'obsolescence, la comparaison s'impose : le réseau Minitel est, malgré le fait qu'il ne fonctionnait pas de la même manière qu'Internet, considéré comme un « ancêtre » d'Internet. On a d'ailleurs pu constater l'arrêt de ce service cette année (jusqu'à cette date, les possesseurs de vieux Minitel pouvaient encore se connecter au réseau). Enfin, la machine en elle-même a un design daté, elle est entrée dans la culture commune comme étant « rétro ».

FIGURE 1.5: Page minitel de France Télécom



Ces deux facettes de la ligne de commande n'en donnent pas une image très reluisante : la ligne de commande serait ou un outil obsolète, ou un signe de « mauvaise santé » de l'ordinateur. Nous ne pouvons pas croire que c'est en ayant ces images en tête que l'on en vient à utiliser la ligne de commande, et surtout, à apprécier cet outil. Il y a donc autre chose.



La ligne de commande : un usage marginal de l'informatique

Les deux réactions-type que nous n'avons pas encore traitées concernent l'illégalité et/ou la complexité de l'activité pratiquée en ligne de commande. Nous avons voulu les traiter séparément et de manière plus approfondie, afin de bien voir en quoi les images auxquelles renvoient ces remarques peuvent être ambivalentes.

Dans l'imaginaire populaire, l'expertise en informatique est souvent associée à l'image du *hacker*, souvent représenté comme un cybercriminel, tapant frénétiquement au clavier des commandes cryptiques en blanc sur fond noir. Sur Internet, on retrouve des images du type de la *figure 1.6*, que ce soit dans la presse ou sur les blogs. La *figure 1.6* représente un homme au visage masqué à la façon des cambrioleurs, avec une carte de crédit en fond, traversée par des suites de 0 et de 1 en blanc, sur fond noir. Tout y est : le contraste qui caractérise la ligne de commande, les suites de zéros et de un, signes à la fois du lien avec l'informatique (qui fonctionne sur une base de chiffres binaires) et de la complexité des commandes, et enfin le couple homme masqué-carte de crédit, qui suggère le vol d'argent sur un compte bancaire. Ces codes reviennent souvent pour évoquer le piratage : la ligne de commande est clairement associée à une activité criminelle.

Il en est de même du cinéma, avec par exemple des images comme les *figures 1.3* et *1.7*, où l'on voit la ligne de commande dans le cadre d'une utilisation experte de l'informatique. Elle est mise en valeur comme telle, comme l'indique le deuxième visuel, où l'écran où les commandes sont affichées est le centre de l'attention : il n'y a que lui à l'image.

Autant de représentations de la ligne de commande dans le contexte d'une activité illégale (le piratage de sites Web) ou experte (lorsque les petits génies de *Sneakers* manipulent la machine). La ligne de commande, nous allons le voir, ne s'utilise pas uniquement dans ces contextes-là, mais la représentation récurrente de cette interface dans ces contextes précis a biaisé notre perception de l'objet.

Lors d'un entretien, l'une des personnes interrogées évoque une histoire qu'il a lue sur Reddit⁴ : « *le mec dans un café, il ouvre un terminal pour faire quelque chose de*

4. Site Web où les utilisateurs peuvent partager des articles sur l'actualité. Il existe plusieurs thématiques, dont la thématique « *commandline* » (ligne de commande, en anglais) centrée sur la ligne de commande.



FIGURE 1.6: Illustration d'un article sur le blog *nanojo.wordpress.com* traitant d'un hacker saoudien qui aurait piraté plusieurs sites.



FIGURE 1.7: Image extraite du film *Sneakers*



tout simple, et il y avait un policier derrière lui. Il le menotte et tout. Il était gêné quand il a compris qu'il n'y avait rien. »

Cette anecdote est éloquente : elle montre à quel point la ligne de commande projette sur son utilisateur une image de pirate informatique, celui qui peut prendre le pouvoir sur un système, aussi complexe soit-il, à des fins criminelles. Le pouvoir du signe est tel que sans même voir ce que l'homme faisait sur ce terminal, le policier l'a immédiatement menotté. C'est dire la peur que la seule apparence de l'interface (le texte blanc sur fond noir, le curseur...) a pu susciter. L'objet technique est ici institué en symbole.

« Dans l'imaginaire populaire, on associe la ligne de commande à faire des trucs dangereux. Au final on a peur des gens qui savent utiliser la technologie. », remarque notre interlocuteur en commentant cette anecdote. Nous sommes bien en face d'une réaction classique devant l'inconnu : on en a peur. C'est en effet certainement cela, le



moteur de cette peur : si le policier avait su que cet homme ne faisait rien de mal, il n'aurait pas réagi. Là, si la seule interface vaut d'être menotté, l'on voit bien que c'est parce que cette pratique de l'informatique est peu courante et chargée d'imaginaires plutôt négatifs (liés à des activités criminelles, ou tout simplement à un pouvoir de l'informatique dont on se méfie) que le policier, dans le doute, a pensé que l'activité de l'homme au café était illégale.

Autre exemple : le héros de *Wargames* est un jeune étudiant « comme tout le monde », « accessible », qui décide de se mettre à « hacker » pour impressionner une fille. Mis à part ses compétences en informatique qui le démarquent du reste des étudiants, il reste un héros proche de nous. C'est « juste un gamin qui voulait tester un nouveau jeu ». Ceci dit, il se retrouve tout de même à pénétrer le programme d'un lance-missile d'Etat, ce qui d'une part est illégal, et d'autre part nécessite un niveau technique nettement au dessus de la moyenne. Nous noterons que l'impact qu'a eu ce film étant assez important (dans l'article de *Wired*⁵ qui lui est consacré, plusieurs remarques sont faites à propos de l'influence qu'a eue le film après sa sortie), il a dû largement contribuer à cet imaginaire dont nous faisons état.

Dans le film *Sneakers*, nous avons un cas similaire : Cosmo, le partenaire de Martin (le personnage principal) est en prison à cause des activités informatiques peu orthodoxes qu'il pratiquait avec son ami.

De manière plus générale, c'est l'imaginaire du « nerd », le jeune homme taciturne, se consacrant uniquement à son ordinateur, qui est mobilisé : celui qui utilise la ligne de commande n'a pas une utilisation de l'informatique « comme tout le monde », il est différent. Non seulement il ne se mêle pas aux autres, mais on ne comprend pas ce qu'il fait. On peut considérer que c'est cet image-là qui est sollicitée dans *Tron*, où on a affaire à un programmeur qui a été mis à pied. Pour retrouver son poste, il passe ses nuits devant son ordinateur à essayer de pénétrer le système de son ex-entreprise, exactement comme un jeune étudiant qui passerait ses soirées à jouer, volets fermés. Plague, un personnage de *Millénium*, nourrit également cette image : c'est un garçon complètement reclus chez lui, obèse, sale, dans un appartement en désordre complet.

5. BROWN, Scott, « WarGames : A Look Back at the Film That Turned Geeks and Phreaks Into Stars », *Wired*, [disponible en ligne : http://www.wired.com/entertainment/hollywood/magazine/16-08/ff_wargames/], publié le 21 juillet 2008, consulté le 31 août 2012



Il est complètement en dehors de toute sociabilité et en rejette les principes, comme celui de garder son appartement et sa personne propres.

Nous remarquerons ici le très faible pourcentage de femmes représentées dans les films que nous avons évoqués. La ligne de commande serait-elle une « affaire d'hommes » ? Les femmes sont rarement aux commandes, elles peuvent être des compagnes, comme dans *Wargames*, mais ne participent pas activement. Les seuls personnages vraiment saillants dans le corpus sur ce point sont Trinity dans *Matrix* et Lisbeth Salander dans *Millénium*. Le fait qu'elles soient des femmes qui prennent l'outil informatique en main les marginalise déjà, parce qu'elles sont vues comme des filles « pas comme les autres » qui font ce que les hommes d'habitude, mais elles bénéficient d'un statut particulier, de quelque chose en plus qui fait qu'elles peuvent elles aussi manipuler la ligne de commande. On ne s'étonne donc pas de l'allure déjà très décalée de Lisbeth Salander, qui porte des piercings partout, des tatouages, est très taciturne, ni du statut d'« élue » dont jouit Trinity. De même avec le personnage féminin de NCIS, gothique et experte en à peu près tout, qui est identifiée comme étant la « nerd » de l'équipe. Elles sont, au final, doublement marginalisées : en tant que femmes et par rapport aux autres personnages du film.

Il semble que la ligne de commande soit donc reléguée, dans l'imaginaire commun, au ban de la société : utilisée par des personnes globalement peu fréquentables, c'est un outil qui provoque, si ce n'est pas de la peur, au moins de la méfiance. Mais nous avons parlé d'ambivalence dans les imaginaires. Il y a effectivement quelque chose qui attire dans cet outil : il est lié, très nettement, à l'expertise en informatique.

L'apanage de l'expert

On voit bien avec *Matrix*, où Neo et Trinity sont des personnages exceptionnels (et donc marginalisés), comment cette marginalité peut devenir quelque chose de très positif : la ligne de commande devient l'apanage de l'expert, celui que l'on écoute, qui sait, qui apporte des solutions.

A ce titre, l'expert impose le respect et l'admiration. C'est cela qui est en jeu dans le « oh tu dois faire quelque chose de compliqué » qu'on entend souvent. On voit très bien comment l'utilisateur de ligne de commande joue ce rôle d'expert dans le film *Wargames*. La jeune fille qui accompagne le héros est impressionnée par ses prouesses



en informatique –dont il lui fait la démonstration dans le but de la séduire, tout de même.

Dans le même ordre d'idées, le titre de *Sneakers* (qui associe plus que clairement la ligne de commande à l'activité des ses héros) est révélateur : en français, il a été traduit par *Les experts*, et présente en effet des personnages clairement identifiés comme étant les meilleurs dans leur domaine. De même que dans *Millénium*, en ce qui concerne Plague et Lisbeth.

Ce qui suscite notre intérêt pour ces films est le fait que non seulement l'utilisation de la ligne de commande en soi est réservé à « celui qui sait » (nous avons souvent des scènes où le personnage est seul ou en compagnie d'observateurs silencieux, et dont on suit les opérations), mais nous y voyons également les autres personnages se positionner par rapport à cela, et afficher de l'admiration, du respect, voire de la méfiance (dans le cas de Lisbeth). Lisbeth est d'ailleurs un excellent exemple : Michael Blomkvist, son coéquipier, est gêné par l'allure pour le moins décalée de la jeune femme, mais la respecte et l'engage parce qu'elle possède les compétences en informatique dont il a besoin pour poursuivre son enquête.

Nous n'irons pas jusqu'à dire que cela se passe exactement comme cela dans la vie réelle, mais ces images, la manière dont elles sont construites par les différents réalisateurs, nourrissent de toute évidence un imaginaire.

Nous insistons sur le fait qu'il s'agit bien d'un imaginaire et que celui-ci est parfaitement construit : « *I cringed during the part in Hackers (1995) when a screen saver with extruded "equations" is used to signify that the hacker has reached some sort of neural flow or ambiguous destination. I cringed for Swordfish and Jurassic Park as well. I cheered when Trinity in The Matrix used nmap⁶ and ssh⁷ (and so did you). Then I cringed again when I saw that inevitably, Hollywood had decided that nmap was the thing to use for all its hacker scenes (see Bourne Ultimatum, Die Hard 4, Girl with Dragon Tattoo, The Listening, 13 : Game of Death, Battle Royale, Broken Saints, and on and on).* »⁸ » écrit le designer de l'interface de

6. Commande permettant de connaître les ports ouverts et les services hébergés sur un serveur.

7. Protocole permettant de se connecter à une machine distante de manière sécurisée.

8. *J'ai grincé des dents pendant la scène de Hackers (1995) quand un économiseur d'écran avec des « équations » extrudées est utilisé pour signifier que le pirate a atteint une sorte de flux de neurones ou de destination ambiguë. J'ai grincé des dents pour Swordfish et Jurassic Park aussi. J'ai applaudi quand Trinity dans Matrix utilisait nmap et ssh (et vous avez fait pareil). Ensuite, j'ai grincé des dents à nouveau quand j'ai vu que, inévitablement, Hollywood avait décidé que nmap était la chose à utiliser pour toutes ses scènes de pirates informatiques (voir Bourne Ultimatum, Die Hard 4, Girl with Dragon Tattoo, The Listening, 13 : Games of Death, Battle Royale, Broken Saints, et ainsi de suite).*

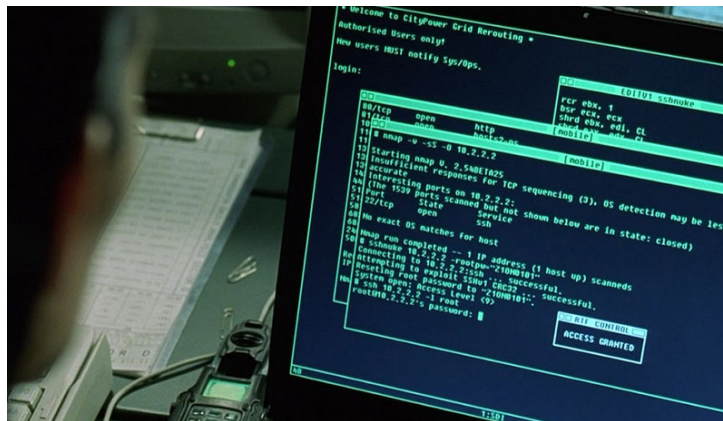


Tron sur son blog

footnoteNIMOY, Joshua T., « Tron Legacy » : [disponible en ligne : <http://jtnimoy.net/workviewer.php?q=178>], publié en 2010, consulté le 10 juin 2012. Il pointe du doigt ici la manière dont l'industrie du cinéma construit l'imaginaire du *hacker*, ou de l'expert en informatique : l'on va choisir une commande Unix susceptible d'impressionner le public, et cette commande sera utilisée par les acteurs lors des scènes de confrontation avec la machine, puisqu'évidemment les acteurs ne sont pas ingénieurs en informatique (Matthew Broderick a même dû apprendre à taper au clavier pour *Wargames*).

Dans le même ordre d'idées, notons que la presse contribue à construire cette image d'expert : dans un article du journal *20 Minutes*, on peut lire l'interview d'un *hacker* qui « donne des conseils dans le monde entier, y compris au FBI », illustré avec la photo en *figure 1.9*⁹. Cette image est tout à fait construite : il est assez évident que le jeune homme a lancé une commande un peu impressionnante pour que la journaliste puisse prendre son ordinateur en photo.

FIGURE 1.8: Trinity, lançant une véritable commande *nmap* dans *Matrix*



Évidemment, les ingénieurs en informatique savent reconnaître la commande et voient bien que c'est de la poudre aux yeux. Mais, ceci dit, aussi artificiel que ce soit, cela semble fonctionner. Au point qu'on attribue –même en plaisantant– l'expertise au fait de faire « comme dans les films », comme en atteste cette remarque, prononcée sur le ton de la plaisanterie lors d'un entretien : « *Un expert informatique connaît aussi*

9. Photo : A. Gelebart/20 Minutes

Article : LARENT, Annabelle, « Matthieu Suiche : "Le hacker, c'est un gentil!" », [disponible en ligne : <http://www.20minutes.fr/high-tech/916317-matthieu-suiche-le-hacker-gentil>], publié le 13 avril 2012, consulté le 31 août 2012



FIGURE 1.9: Illustration d'un article sur le site de *20minutes.fr* comportant l'interview d'un *hacker*

bien l'architecture hardware ET software et est bien sûr une star en réseau et manipule tout en ligne de commande pour faire comme dans les films. ».

Lors d'un entretien, nous avons appris qu'il existe des jeunes qui ont compris ce que la ligne de commande faisait rejaillir sur leur personne cette expertise, et qui apprennent quelques commandes pour avoir l'air très forts en informatique : « *Le Kikoo, c'est Jean-Kévin. Ça vient des mecs qui ponctuent leur phrases par LOL, font kikoo lol aso. [...] Il frime parce qu'il sait utiliser trois commandes mais ne sait pas vraiment [s'en servir].* » De la même manière que les connaisseurs prennent de la distance avec les « *hacker scenes* » des films que nous avons évoqués, ils se moquent également des « *Kikoo* » qui utilisent ostensiblement la ligne de commande pour avoir l'air intelligents. Mais les personnes de leur entourage qui ne s'y connaissent pas tant que cela sont susceptibles de tomber dans le panneau. C'est ainsi que lors d'un entretien, notre interlocuteur nous avoue qu'il a « *gagné en crédibilité dans [son] entourage [...] genre le mec il utilise la CLI, c'est un bonhomme de l'info, vaut mieux le croire, dans le doute* ».

Nous ne devons pas qu'à ces films et à ces jeunes gens la construction et la réactivation de l'imaginaire de l'expertise. Il nous faut aussi remarquer que le quotidien de la plupart des personnes possédant un ordinateur y contribue également. Aujourd'hui, la plupart des gens travaillent avec une interface graphique. Les personnes qu'ils appellent lorsqu'ils ont un problème avec leur ordinateur leur conseillent en général d'ouvrir un terminal ou une invite de commande (l'invite de commande DOS pour Windows, le Terminal pour Mac) pour y entrer une commande, parce qu'en général, il est plus simple de régler un problème informatique à distance grâce à ces outils.



De ce fait, beaucoup de gens associent le ligne de commande à un certain degré d'expertise en informatique, puisqu'il ne la rencontrent que lorsqu'ils font appel à un « expert », justement.

C'est d'ailleurs aussi parce que les machines encore totalement administrées *via* la ligne de commande sont des serveurs (sur lesquels on n'installe pas d'interface graphique) administrés par des spécialistes (les administrateurs système), que la ligne de commande évoque autant l'expertise. C'est un outil perçu comme dévolu à des activités sérieuses, à laquelle n'importe qui ne peut pas s'adonner. C'est assez palpable dans le discours des novices sur les forums de Ubuntu France, à qui on conseille une commande pour régler le problème qu'ils rencontrent, et qui avouent avoir peur que cela « casse tout », ou « soit dangereux ». On rencontre également des remarques telles que : « *Ubuntu, ça a l'air bien, mais il faut s'y connaître en informatique, on m'a dit qu'il fallait maîtriser la ligne de commande.* » sur les forums, de la part de novices.

Cet ensemble d'images et de discours fait, selon nous, de la ligne de commande la boîte à outil de l'expert, qu'on respecte, mais dont on est tenté d'envier le pouvoir sur les machines – et celui des machines, qu'il contrôle. Ce n'est donc pas étonnant que l'on aie une ligne de commande qui soit perçue comme un outil « puissant » qui permet d'exploiter toutes les performances de la machine, puisque ceux qui l'utilisent sont investis de ce pouvoir.

Il ressort de ce tour d'horizon que la ligne de commande semble associée à l'illégalité, à l'expertise, et, surtout et de manière plus générale, elle semble être le signe de l'appartenance à une certaine marginalité. Celle de leurs utilisateurs, que l'imagerie cinématographique représente souvent comme taciturnes, peu fréquentables, mais aussi celle de la pratique de l'informatique.

Ces personnages provoquent autant l'attraction que la répulsion, nous avons bien un imaginaire ambivalent :

- *attraction*, parce c'est l'outil réservé aux « meilleurs »,
- *répulsion* : on a parlé de marginalité, de criminalité. Celui qui utilise la ligne de commande, c'est aussi celui qui va détourner de l'argent à la banque, qui va usurper votre identité...



Les imaginaires rattrapés par la réalité

Nous sommes cependant plongés dans l'étude d'un imaginaire socio-technique qui nous donne une vision tronquée de l'objet : cet outil sert aussi au quotidien à utiliser un ordinateur, sans que cela soit dangereux ou particulièrement compliqué. Les entretiens soulignent ce décalage entre la réalité de leur activité (éditer du code sur leur machine, parler sur IRC... et ce que l'observateur est amené à penser. Les cours d'initiation à la ligne de commande que nous avons pu consulter¹⁰ sont, de plus, loin d'être des manuels de piratage : on y apprend comment changer de répertoire, comment déplacer ou copier un fichier, comment convertir une image, en somme des opérations très simples, que l'on est amené à faire de manière quotidienne. Il y a des gens qui s'en tiennent à ce type de compétence sans jamais aller plus loin.

La première chose que nous pouvons remarquer est qu'une grande partie de la filmographie ayant marqué la « culture geek », comme *Wargames*, sont des films datés, où l'on utilise une interface en ligne de commande parce que c'était la seule interface disponible à l'époque. Comme ce film a marqué sa génération (au point que Google décide d'organiser une projection du film pour le ving-cinquième anniversaire de ce dernier *Wargames*¹¹), ceux qui ont vu le film gardent en tête cette image-là de l'informatique difficile. Mais aujourd'hui, l'informatique a changé : les ordinateurs ne prennent plus, pour commencer, la place folle qu'ils prennent dans le film...

L'autre chose que nous pouvons remarquer, toujours au regard de cette filmographie, c'est que les scénarios ex-mêmes relèvent ou de situations à caractère exceptionnel (*Wargames* relate l'histoire d'un étudiant qui pénètre le programme d'un lance-missiles d'Etat, ce qui n'advient pas tous les jours), ou des situations relevant de la science fiction (*Matrix* et *Tron*).

Rappelons également que nous parlons ici d'un imaginaire essentiellement véhiculé par le « signe » ligne de commande, et non sa réelle utilisation : dans *Wargames*, l'opération que le jeune hacker effectue pour rentrer dans l'ordinateur est beaucoup plus complexe que ce qu'on le voit faire à l'écran. Dans un article de *Wired* traitant

10. Un extrait d'un des cours est disponible en annexe, p.93

11. BROWN, Scott, « WarGames : A Look Back at the Film That Turned Geeks and Phreaks Into Stars », *Wired*, [disponible en ligne : http://www.wired.com/entertainment/hollywood/magazine/16-08/ff_wargames/], publié le 21 juillet 2008, consulté le 31 août 2012



de l'héritage laissé par le film *Wargames*¹², les réalisateurs expliquent comment les scènes au centre desquelles il y avait une manipulation informatique complexe (en ligne de commande) ont été conçues. Malgré la présence de véritables informaticiens compétents sur place, dont on avait demandé l'oeil vigilant pour « faire vrai » le plus possible, les réalisateurs ont préféré filmer et monter la scène comme si cela était très rapide, alors que les manipulations –qui existent réellement– nécessaires au héros pour prendre le contrôle de la machine qu'il vise, par exemple, sont beaucoup plus complexes et longues que ce que l'on a pu voir à l'écran.

« *You could get all the hacker geekiness you wanted just by standing on the set. We were dealing with things like when Matthew sits at the computer, we've got an actor who can't even type. I'd say, "No, I just really want him to type in 'David' and have him get on."* They said, "No! You can't do that! You have to go through all these elaborate sequences!" I said, "No, we're not doing that. Audiences will have left the theater by the time he logs into the computer one time."¹³ » raconte John Badham, le réalisateur, dans l'article.

L'imaginaire généré par ce film culte (et par les discours à propos de la ligne de commande qui viennent s'y rajouter) est donc basé sur des images déjà artificielles, déjà construites, façonnées par la vision que le réalisateur a de l'informatique et qu'il veut présenter de celle-ci. Nous ne pouvons pas en rester là.

1.2 Un discours d'escorte euphorique

Un outil puissant et proche de la machine

Nous venons d'analyser les discours de personnes ne connaissant rien à la ligne de commande : on comprend leur appréhension, due au fait qu'ils ne connaissent pas l'outil. Mais qu'en est-il de ceux qui l'utilisent ?

12. BROWN, Scott, « WarGames : A Look Back at the Film That Turned Geeks and Phreaks Into Stars », *Wired*, [disponible en ligne : http://www.wired.com/entertainment/hollywood/magazine/16-08/ff_wargames/], publié le 21 juillet 2008, consulté le 31 août 2012

13. *Vous pouviez obtenir toute la « geekitude » du hacker dont vous aviez besoin en étant seulement sur le plateau. Nous devons nous débrouiller avec des choses comme quand Matthew s'assied devant l'ordinateur, nous avons un acteur qui ne sait même pas taper à la machine. Je disais : "Non, je veux vraiment qu'il tape seulement 'David' et qu'il se connecte". Ils disaient : "Non! Vous ne pouvez pas faire ça! Vous devez passer par toutes ces séquences élaborées!". Je disais : "Non, nous n'allons pas faire ça. Le public aura quitté la salle, le temps qu'il se connecte à l'ordinateur."*



Ce que nous remarquons d'emblée, c'est l'euphorie dont sont emprunts tous les témoignages que nous rencontrons. Les utilisateurs de la ligne de commande sont convaincus d'utiliser ce qu'il y a de mieux en informatique : un outil puissant, modulable, proche de la machine. . Ils ne tarissent pas d'éloges. Ils vont même jusqu'à revendiquer leur choix : dans un article de *Wired*¹⁴, on apprend que suite à un post de blog expliquant en quoi la ligne de commande devait laisser le pas aux interfaces graphiques, de nombreux commentateurs sont venus « défendre » la ligne de commande.

Cela va plus loin que la défense de son choix d'interface dans le cadre d'un débat : sur la page Reddit dédiée à la ligne de commande¹⁵, l'on peut trouver un lien vers une page intitulée « *10 Reasons Why the Command Line is More User-Friendly than the Desktop*¹⁶ ». La page a été traduite en plusieurs langues, et de nombreux commentateurs sont venus l'appuyer.

L'article se présente comme une liste des avantages de la ligne de commande vis à vis des interfaces graphiques. Il rassemble les remarques que l'on peut entendre par ailleurs, notamment lors d'entretiens (la ligne de commande est standard, il est plus facile de s'y retrouver alors que les interfaces graphiques ne le sont pas, par exemple). Nous n'allons pas toutes les énumérer ici, ce qui nous intéresse est d'abord le geste de lister les qualités de la CLI que les arguments avancés. Soulignons que l'on ne trouve pas aussi facilement de liste de ce type vantant les bienfaits des GUI. Nous avons pu dénicher une page assez complète¹⁷, qui met un peu en avant l'interface graphique, mais toujours en comparaison avec la CLI. Il n'y a pas de liste spécifique comme celle que nous avons citée.

La comparaison entre les deux pages web est assez éloquente : les deux se présentent comme des comparaisons entre les deux interfaces, mais la première est centrée sur la ligne de commande, tandis que la seconde présente les deux interfaces l'une à côté de l'autre, en indiquant les défauts et les qualités de chacune. Les utilisateurs de ligne de commande sont-ils si objectifs que cela ?

14. GARLING, Caleb, « Why the GUI Will Never Kill the Sacred Command Line », *Wired Enterprise*, [disponible en ligne : <http://www.wired.com/wiredenterprise/2012/07/command-line>], publié le 10 juillet 2012, consulté le 27 juillet 2012

15. <http://www.reddit.com/r/commandline>, consulté le 25/08/2012

16. disponible en annexe, p.97

17. « Comamnd line vs. GUI », *Computer Hope*, [disponible en ligne : <http://www.computerhope.com/issues/ch000619.htm>], consulté le 20 août 2012



Nous pouvons en douter. Ce positionnement centré sur la ligne de commande apparaît très clairement dans l'article de *Wired*¹⁸, puisqu'il donne la parole aux tenants des GUI et de la CLI. On y lit que la ligne de commande possède des défauts, est parfois limitée pour certains usages. . . ce que gomme totalement la première liste que nous avons évoquée.

La plupart des points soulevés par les deux pages sont des arguments tout à fait recevables. Par exemple, le fait que la ligne de commande facilite le don et la réception de conseils pour débloquer une information se vérifie : il suffit de donner à la personne en besoin la commande idoine, et de copier le message d'erreur produit par le programme (la plupart des applications en ligne de commande sont versatiles), pour obtenir de l'aide : les forums Ubuntu en sont la preuve. Nombreux sont les échanges ponctués de commandes et de messages d'erreur dans les sujets ayant trait à l'installation ou à la configuration. Mais encore une fois, ce qui nous intéresse ici n'est pas de savoir qui a raison, puisque, comme le souligne l'article de *Wired*, le débat semble intarissable, et que nous sommes concentrés sur la manière dont les arguments sont amenés.

Penchons-nous sur l'un de ces arguments, celui de la proximité avec la machine. Il consiste à dire que la ligne de commande donne un meilleur accès à des « couches » inférieures de la machine, plus proches de l'architecture système. Cet argument est notamment avancé dans l'essai *In the beginning was the command line*¹⁹, qui compare les GUI à des éléments cosmétiques. Nous en avons débattu lors d'un entretien :

« C'est le côté rassurant du WYSIWYG²⁰, c'est l'impression que tu obtiens en temps réel tes modifications. Or un ordinateur aujourd'hui c'est du temps partagé. Entre le moment où t'a cliqué et l'action, il y a des dizaines des cycles processeur. Et en ligne de commande, c'est un cycle. Et ça c'est une certitude. [. . .]

En réponse à une remarque évoquant des tenants de la ligne de commande qui critiquent l'interface graphique en disant : « tu ne vois pas ce que la machine fait » :

18. GARLING, Caleb, « Why the GUI Will Never Kill the Sacred Command Line », *Wired Enterprise*, [disponible en ligne : <http://www.wired.com/wiredenterprise/2012/07/command-line>], publié le 10 juillet 2012, consulté le 27 juillet 2012

19. STEPHENSON, Neal, *In the beginning was the command line*, 1999

20. What You See Is What You Get, interface comme celle du traitement de texte, où ce qui s'affiche à l'écran est identique au résultat final



« parce que souvent les gens qui utilisent la ligne de commande ont mené une réflexion sur l'utilisation de l'informatique, je pense. »

Nous notons ici que notre interlocuteur reconnaît que cet argument est valable si l'on a « mené une réflexion sur l'utilisation de l'informatique », ce qui est le cas des utilisateurs de la ligne de commande, surtout s'ils sont issus de formations en informatique. Cela rend donc l'argument inaccessible à la majorité des gens, qui utilisent des interfaces graphiques sans se poser plus de questions que cela. Nous sommes toujours en train de regarder la ligne de commande du point de vue de ceux qui l'utilisent.

Peut-être avons-nous affaire à des personnes très enthousiastes à propos de leur outil, ce qui biaise notre regard. Le fait qu'un discours similaire soit tenu dans des manuels d'initiation à la ligne de commande nous invite à penser qu'il s'agit bien d'un discours d'escorte, un discours qui accompagne la ligne de commande, et l'investit d'un pouvoir, d'une efficacité particulière, en gommant ses défauts. Dans le manuel *Introduction to the command line*²¹ le premier chapitre, intitulé « Put Yourself In Command »²² est consacré à rappeler combien la ligne de commande est un outil plus efficace que l'interface graphique : « *Commands enable you to do all sorts of powerful things on your computer. We'll demonstrate this by looking at an everyday task that might be familiar to you.* »²³ ».

Si nous sommes en train de lire ce manuel, c'est que nous sommes à priori intéressés par la ligne de commande, mais l'auteur a tout de même jugé bon de nous introduire à l'apprentissage de la ligne de commande par ce biais : il s'agit vraiment de répéter un discours positif pour donner une valeur à l'objet.

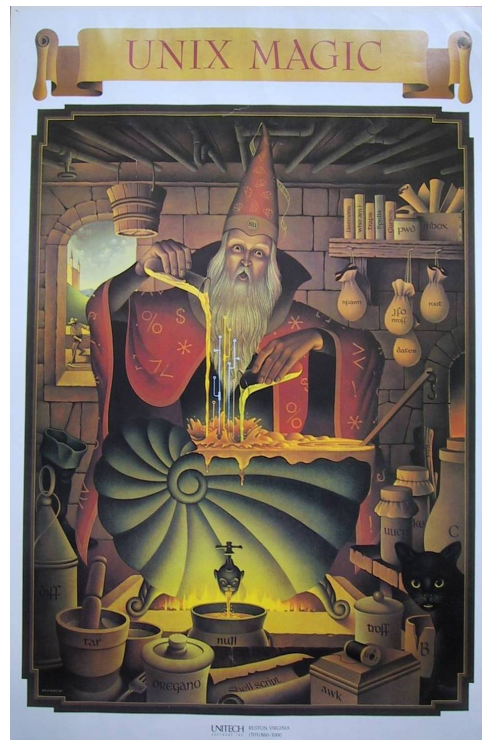
Terminons notre exploration des discours sur la ligne de commande par cette image, trouvée sur Internet au hasard d'une recherche, qui résume bien la manière dont les utilisateurs de la ligne de commande perçoivent leur outil : ils ont entre les mains un outil tellement puissant qu'ils sont comme des magiciens à combiner les commandes entre elles (chaque récipient sur l'image porte le nom d'une commande Unix connue).

21. HYDE, Adam, *Introduction to the command line*, FLOSS Manuals, 2009

22. Extrait disponible en annexe, p.90

23. *Les commandes vous permettent de faire toutes sortes de choses puissantes avec votre ordinateur. Nous allons le démontrer en regardant une tâche quotidienne qui devrait vous être familière.*





Ainsi, nonobstant le bien-fondé de beaucoup d'arguments, il semble manifeste que les utilisateurs de la ligne de commande sont les porteurs d'un discours d'escorte euphorique à propos de leur outil –un discours qui se place sur un tout autre registre que celui des imaginaires que nous avons étudiés, centré sur l'efficacité de l'outil. En effet, le geste qui consiste à dresser la liste des avantages de la CLI est déjà le signe que le discours dont nous sommes les témoins est biaisé, construit. Comme tout discours d'escorte, il a pour but d'investir notre objet d'une valeur particulière, dont ce dernier n'est pas forcément pourvu au départ.

L'on voit bien ici que les difficultés que l'on peut rencontrer à l'apprentissage, puis à l'utilisation quotidienne de la ligne de commande sont gommées : le discours n'est pas objectif, il ne montre que les « bons côtés » de la CLI. Serait-ce la nostalgie qui jouerait le rôle de cosmétique ?



C'était (vraiment) mieux « avant » ?

« *The command line is a crusty, old-fashioned way to interact with a computer, made obsolete by GUIs, but a small hardcore of people who refuse to move on still use it.*²⁴ » écrit un dénommé MSP sur le site TuxRadar, d'après *Wired*²⁵.

C'est en effet ce que les détracteurs de la ligne de commande opposent souvent comme argument à ceux qui l'utilisent. On serait tenter de penser que ces derniers continuent à utiliser la ligne de commande par nostalgie. Est-ce vraiment le cas ?

Notons d'emblée que la filmographie en lien avec l'informatique –dont nous n'avons exploré qu'une petite partie– a eu une certaine influence, et l'on peut aisément déduire que cette influence a dû rendre plus d'un nostalgique, puisque ces films sont datés. De même, le livre *Hackers* de Steven Levy, en relatant avec enthousiasme le parcours des premiers *hackers* du MIT, met en scène les machines de l'époque de manière presque attachante. Il raconte notamment comment le vieux PDP-6 a marqué les *hackers*²⁶ : « *I couldn't imagine being without a PDP-6*²⁷ » confesse l'un d'eux.

Une nostalgie que l'on voit poindre dans le fait que la ligne de commande sous GNU/Linux est quasi-systématiquement par défaut en blanc sur noir : ce choix de couleur vient en réalité de l'époque où sont apparus les premiers ordinateurs dotés d'écrans, pour lesquels on a choisi l'affichage le plus économique : la couleur verte était la moins demandeuse en énergie. Aujourd'hui, nos ordinateurs sont plus puissants, nous pouvons tout afficher en noir sur blanc, comme sur une feuille de papier –les terminaux sur Mac OS sont d'ailleurs ainsi– mais l'interface est restée la même. Pour les terminaux modernes, on a seulement remplacé le vert par du blanc, plus agréable à la lecture.

On voit également poindre de la nostalgie dans la manière dont les programmes eux-mêmes fonctionnent. L'éditeur de texte en ligne de commande *Vim* est par exemple basé sur deux modes principaux : le mode « visuel » et le mode « édition », fonctionnement qui a été soigneusement évité par les designers de Xerox PARC

24. *La ligne de commande est une manière d'interagir avec un ordinateur, compliquée, passée de mode, rendue obsolète par les GUIs, mais un petit groupe d'irréductibles refuse toujours de l'utiliser.*

25. GARLING, Caleb, « Why the GUI Will Never Kill the Sacred Command Line », *Wired Enterprise*, [disponible en ligne : <http://www.wired.com/wiredenterprise/2012/07/command-line>], publié le 10 juillet 2012, consulté le 27 juillet 2012

26. LEVY, Steven, *Hackers : Heroes of the computer revolution*, Sebastopol CA, O'Reilly Media, 2010, p.78

27. *Je ne ne pouvais pas m'imaginer être sans un PDP-6*



lorsqu'ils ont conçu leur interface²⁸. Les modes sont obsolètes, puisqu'ils ont été très vite écartés par les concepteurs d'interface et qu'aujourd'hui, les interfaces les plus récentes ne fonctionnent plus avec des modes. Malgré cela, un grand nombre de personnes utilisent Vim pour écrire du code, par exemple. C'est un programme très répandu, au point d'être parfois livré par défaut avec certaines distributions GNU/Linux.

De même, la mise en forme des mails en ligne de commande est datée : « *Ce qui est rigolo, c'est qu'on a des restrictions qui n'ont plus lieu d'être dans le terminal. On continue à envoyer des mails cassés à 72 caractères alors que les écrans ont grandi.* » raconte une personne interrogée lors d'un entretien. Le programme a gardé cette mise en forme alors que le matériel aujourd'hui permet de s'en débarrasser, comme s'il s'était figé dans le temps.

Il est vrai que la ligne de commande étant un outil daté (mais pas le plus ancien de l'histoire de l'informatique, puisque les ordinateurs fonctionnaient auparavant avec des cartes perforées !) et qu'il n'a pas beaucoup évolué. Mais, lorsqu'on se penche sur les interfaces graphiques, il n'est pas sûr qu'elles aient tant évolué que cela : « *Look at a screenshot of the original Macintosh and compare it with one of the latest versions of OSX. With the exception of some new gadgets and some smoke and mirrors, much of the interface is still the same.* »²⁹

Peut-on donc juger l'innovation en informatique seulement à partir des interfaces ? Cela nous semble abusif. D'autant que nous savons que les programmes eux-mêmes sont en constante évolution.

En effet, beaucoup de projets de logiciels libres en ligne de commande sont encore en actif développement : ils sont loin d'être obsolètes. La plupart des distributions Linux sont également régulièrement mises à jour. Remarquons également que *bash* (aussi connu sous le nom de *Bourne-Again Shell*, car il est lui-même une amélioration du *shell* historique d'Unix, le *Bourne shell*), le principal langage utilisé pour la ligne de commande et installé par défaut dans la plupart des distributions GNU/Linux, n'est

28. « When he was at Apple, Larry Tesler had a license plate saying "NO MODES," emphasizing his passion for designing software that would be simple and easy to use. » – B. Moggridge, *Designing Interactions*, The MIT Press, 2006, p.57 (Quand il était à Apple, Larry Tesler avait une plaque qui disait « PAS DE MODES », pour mettre en lumière sa passion pour le design de logiciels qui seraient simples et faciles à utiliser)

29. RASKIN, Aza, « The linguistic Command Line, *Interactions*, vol. 15, n. 3, p. 44-45, 2008 (Regardez une capture d'écran du Macintosh d'origine et comparez-la avec la dernière version d'OSX. À l'exception de quelques effets d'ombre et de miroir, l'interface est toujours la même.)



pas le seul langage que l'on peut utiliser : depuis que celui-ci a été mise en place, de nouveaux sont apparus (*zsh*, par exemple).

Si bien que l'informatique telle que nous l'utilisons il y a une ou deux générations n'a rien à voir avec celle que nous connaissons actuellement, même s'il s'agit du même type d'interface. L'interface en ligne de commande de l'époque était beaucoup moins pratique à utiliser qu'aujourd'hui. Les premiers *shells*, par exemple, ne comportaient pas d'autocomplétion. A l'époque où il n'y avait pas d'interfaces graphiques, il n'y avait pas le choix, mais si aujourd'hui, on fait le choix d'utiliser la ligne de commande, ce n'est pas pour revenir à cela. Il y a fort à parier que si le *shell* n'avait pas évolué, nous n'aurions pas cet engouement pour la ligne de commande.

Il nous semble donc qu'il ne faille pas généraliser : s'il y avait eu une réelle résistance au changement, la ligne de commande n'aurait pas bougé du tout, et les interfaces graphiques auraient continué à évoluer. L'outil qu'ils utilisent est effectivement un héritier de la manière dont on faisait de l'informatique « avant », mais il est trop différent de ce qui se faisait avant pour qu'on puisse ramener l'un à l'autre.

Xerox PARC et l'informatique personnelle

Ceci dit, nous ne pouvons nier que l'interface graphique est intervenue à un moment donné de l'histoire du développement de l'informatique pour supplanter la ligne de commande, ou, à minima, se positionner comme une alternative à cette dernière.

Aujourd'hui, la plupart des ordinateurs circulant sur le marché sont livrés avec des GUIs, qui ont supplanté, dans les années 80, l'interface en ligne de commande. Si ces interfaces ont rencontré le succès que l'on peut constater, c'est que les analyses des chercheurs du Xerox PARC ont été justes.

Ces dernières se basaient sur des utilisateurs vierges de toute connaissance particulière en informatique (ils se posaient régulièrement la question de savoir si leur grand-mère aurait pu utiliser l'interface). La volonté était d'en faire un outil suffisamment simple pour être accessible à n'importe qui. Ce qui n'est pas le cas de la ligne de commande qui nécessite toujours un apprentissage –même minime– au départ.



L'expérience la plus emblématique de cette démarche est celle qui a été faite sur des jeunes filles³⁰, qui n'avaient absolument aucune connaissance en informatique : on les mettait devant un écran vide, et à leur disposition des outils de pointage différents (une souris, un clavier. . .) et on observait leur comportement. L'idée était vraiment de partir de ce que faisait l'utilisateur pour s'adapter à lui. C'est l'une des raisons qui ont fait que les innovations de Xerox PARC ces années-là ont connu le succès que l'on sait : comme les interfaces étaient adaptées aux utilisateurs –et à des utilisateurs sans expérience– elles étaient très accessibles. Il est donc devenu possible de vendre des « ordinateurs personnels » à grande échelle parce que l'on savait que n'importe qui pouvait les prendre en main.

C'est ce que pointe Neal Stephenson dans son essai en remarquant que « *we want GUIs because they are convenient and because they are easy* »³¹. Il admet que les interfaces graphiques se rapprochent de l'utilisateur.

L'interface en ligne de commande demandait un apprentissage trop long pour qu'à une échelle commerciale, cela devienne rentable. On a donc pensé des interfaces qui pouvaient être utilisées sans effort préalable particulier. C'est cela qui a motivé le travail sur la métaphore si important qui a été fait pour mettre au point ces interfaces. Il a fallu trouver des équivalents dans la vie réelle pour les principales opérations pour que les utilisateurs ne soient pas perdus (déplacer un dossier, jeter quelque chose à la poubelle. . .).

Au final, qu'on l'utilise ou pas, les discours et les images que l'on a de la ligne de commande sont tout autant artificiels, construits et décalés avec le réel. Nous avons affaire à des discours d'escorte concurrents, qui ne délivrent pas d'information –ou une information biaisée– sur la réalité de l'outil. Mais nous sommes forcés de constater que ces discours ont un effet. Ils sont nourris et réactivés par de nouvelles images et de nouveaux discours.

30. MOGGRIDGE, Bill, *Designing Interactions*, The MIT Press, Cambridge, 2006, p.27

31. STEPHENSON, Neal, *In the beginning was the command line*, 1999 (*Nous voulons des GUIs parce qu'elles sont pratiques et faciles à utiliser*)



1.3 Des imaginaires artificiels, mais réactivés par les usages

Pris au jeu de l'imaginaire, de manière consciente ou non

Ces imaginaires dont on voit bien qu'ils sont en grande partie construits sont, malgré cela, réactivés en permanence. L'un des ressorts les plus puissants de cette réactivation se trouve au niveau des utilisateurs de ligne de commande eux-mêmes, comme l'indique ce verbatim recueilli en entretien. Ici, cet étudiant sait pertinemment que la ligne de commande crée une réaction de peur auprès de ses camarades, c'est la raison pour laquelle il lance des commandes en public :

« En réalité, je lance la ligne de commande à Sciences Po quand il y a des gens derrière moi pour leur faire peur. Une fois j'étais en amphi, le cours était pas intéressant. Je commence à lire un truc en arabe. Pour lire en arabe je regarde tous les liens... Je lis ça, et après j'ai ouvert mon terminal, et les gens derrière moi ont fait : "qu'est ce que tu fais ?". C'est marrant, je ne vois pas le rapport. Je faisais un truc un peu bizarre pour eux. On leur a jamais montré. »

Autre exemple, tiré d'un autre entretien :

« J'ai lancé un redoutable "du"³² pour son défilement vitesse grand V en vert sur fond noir, ça a fait son petit effet. »

Ce simple geste réactive les peurs liées à la ligne de commande. Remarquons que cette « expérience » est faite de de manière suffisamment fréquente pour qu'une application web soit créée uniquement dans ce but : *Hacker Typer*³³. Tout y est : le nom, la manière dont il est présenté sur Internet (une page d'accueil avec du texte vert sur fond noir), et la manière dont il fonctionne. Cet outil permet de faire semblant de taper très vite des choses très compliquées dans une console. Alors qu'en fait on ne fait qu'ouvrir une page web et taper au hasard sur le clavier. Remarquons qu'il n'est nul besoin de connaissances en informatique pour utiliser cette application : nous pouvons supposer que ce geste de « faire peur aux copains en tapant très vite sur un clavier des choses écrites en vert sur un écran noir » est reproduit par des personnes qui n'utilisent pas la ligne de commande. Autant dire que cet imaginaire

32. Une commande très simple qui permet de voir l'espace disque utilisé sur son ordinateur.

33. « Hacker Typer », [disponible en ligne : <http://www.hackertyper.net>], consulté le 31 août 2012



du dangereux *hacker* est réactivé par tellement de gens qu'il a encore de beaux jours devant lui.

Il est également réactivé dans des discours, comme ce commentaire à propos de l'illustration d'un article de blog : « *Pour les vieux raccros, non l'image n'a aucun rapport mais ça fais hacker la console linux*³⁴ ».

La ligne de commande « *ça fais hacker* » –en citant scrupuleusement l'auteur... Inutile de chercher très loin pour voir que cet imaginaire que nous venons de déconstruire et dont nous voyons bien qu'il n'est pas ancré dans la réalité de l'utilisation de l'informatique, est tout de même réactivé en permanence. La remarque du blogueur citée précédemment est tout à fait révélatrice : il *sait* que l'usage de la ligne de commande n'est pas circonscrit aux *hackers* (et à cette image-là du *hacker*, bien loin de la réalité des hackerspaces), loin s'en faut, mais il sait que son public, lui fait cette association-là, alors il illustre ainsi son blog pour se faire valoir. On remarque aussi une chose, d'ailleurs, c'est qu'il traite ses éventuels détracteurs de « *vieux raccros* », soulignant par là la vétusté de l'outil.

De même avec cette remarque, faite lors d'un entretien : « *Mon terminal c'est Quakee, c'est un drop down, qui se superpose à ton bureau, c'est classe, en plus c'est transparent, ça fait un peu hacker.* »

La réactivation de ces imaginaires se fait souvent de manière totalement assumée. Dans une école d'informatique, pour les portes ouvertes, l'on utilise les mêmes astuces qu'au cinéma (des effets d'interface très « *tape à l'œil* ») pour signifier le bon niveau et faire bonne impression aux visiteurs :

« *Putain j'ai honte. Quand on organise des visites de notre labo pour des lycéens ou plus jeunes, souvent on leur fait une pseudo démo ou on crack un hash sur un cluster [...] Ben j'ai du y mettre plein d'effets débiles a la demande des gens qui organisent ça, genre pendant que l'attaque tourne ça met des mots de passe au hasard qui clignotent [...] Pis pour donner l'impression qu'il se passe quelque chose pendant que l'attaque tourne ca fait des dd au hasard sur les machines pour faire clignoter les leds des disques [...] juste pour faire joli quoi.*³⁵

34. *Iccream*, « Comment hacker le web très très rapidement », Paperblog, [disponible en ligne : <http://www.paperblog.fr/601102/comment-hacker-le-web-tres-tres-rapidement/>], publié le 4 avril 2008, consulté le 31 août 2012

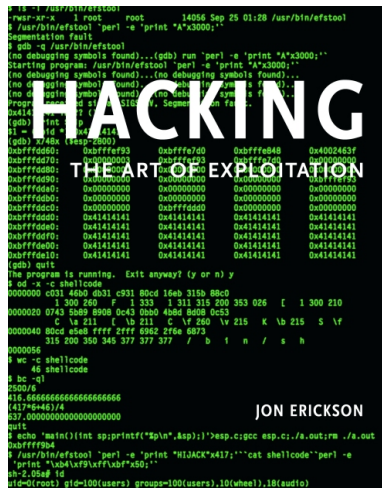
35. le texte original est disponible en annexe (p.118), il a été légèrement mis en forme ici pour rendre la lecture plus aisée



C'est un cas similaire à celui du « kikoo », qui fait le malin avec les trois commandes qu'il connaît, à la différence qu'ici, les personnes qui organisent ces portes ouvertes possèdent un niveau informatique bien supérieur, mais savent qu'ils doivent montrer au public ce qu'il attend, à savoir ce qu'il a vu dans les films. Ils sont pris au jeu d'un imaginaire qu'ils réactivent en présentant leur école ainsi, même s'ils savent que c'est très artificiel.

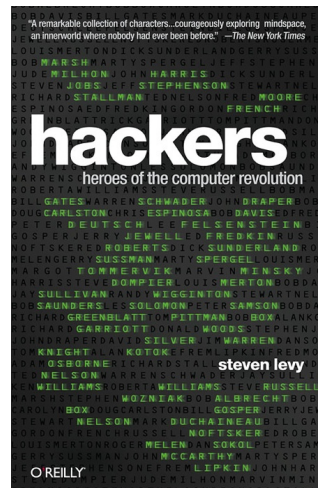
Nous remarquons ce phénomène de réactivation d'imaginaires dans la création d'objets liés à l'informatique : des livres, par exemple.

La couverture du livre *Hacking : the art of exploitation*, qui expose des techniques de piratage informatique, joue là-dessus : du texte vert sur fond noir, des suites de caractères indéchiffrables pour des néophytes.



La couverture de l'ouvrage de Steven Levy, *Hackers : Heroes of the computer revolution* arbore un design du même type : fond noir, texte vert, dans une police type « console ». Ce qu'il est intéressant de relever ici, et que nous n'avons pas avec l'autre couverture de livre, est que cet ouvrage relate l'histoire des *hackers* depuis leur apparition à la fin des années cinquante jusqu'à aujourd'hui, en insistant sur la valeur positive du mot *hacker* : une manière alternative et curieuse de faire de l'informatique. Il ne s'agit pas ici de l'histoire des cybercriminels. La couverture joue pourtant sur cet imaginaire lié à la cybercriminalité, à tout ce qui fait peur dans la ligne de commande, alors que globalement, il ne s'agit pas de cela : l'imagerie populaire a juste été reprise par l'éditeur, arrangée pour les besoins du livre et dans le but de le faire vendre.





Que ce soit de la part d'utilisateurs de la ligne de commande ou de néophytes (dans le cas de *Hacker Typer*), l'image de la ligne de commande comme étant l'apanage de l'expert en informatique, voire du cyber-criminel, est réactivé.

L'exploitation à échelle commerciale de cet imaginaire

Mais il existe un autre levier de réactivation de ces imaginaires : leur utilisation dans un contexte commercial. C'était déjà le cas avec les films que nous avons étudiés. Mais il est encore plus intéressant de se tourner vers des productions récentes pour bien voir le mécanisme de « reprise ».

Premier cas qu'il nous semble intéressant à étudier : la messagerie Gmail de Google. Les utilisateurs de cette dernière ont la possibilité de personnaliser l'affichage des messages en utilisant des « thèmes » prédéfinis par Google. Parmi ceux-ci, est proposé un thème « Terminal », qui initialement, ressemblait à ce que l'on peut voir en *figure 1.10*.

Ce « thème » reprenait tous les codes de l'esthétique de la ligne de commande : le vert sur fond noir, l'ascii art, la police « terminal ».

Au passage à la nouvelle version de Gmail, les designers de Google ont fait ce geste très intéressant de ne pas réintégrer le nouveau thème, mais de l'adapter à la nouvelle version de la messagerie, ils l'ont épuré, pour ne garder que le discret signe du curseur vert qui clignote sur l'écran noir. Ce qui se passe sur Gmail est le même phénomène que l'on rencontre avec *Tron* : la réactivation d'un imaginaire daté de l'informatique.



FIGURE 1.10: L'interface de Gmail en thème « Terminal »



Le premier film *Tron* exploitait l'imaginaire de l'informatique et du futur tel qu'il était à l'époque à laquelle il a été conçu, et le deuxième film est venu réactiver cet imaginaire en y ajoutant de nouveaux éléments. Le premier film *Tron* est marqué par une esthétique très particulière, tout en bleu et noir, qui correspond à tout un réseau d'images, de discours, de projections (notamment cette idée très forte de pouvoir entrer *dans* la machine, que l'on retrouvera avec *Matrix*) qui avaient cours dans les années 1980, mêlé à des choix esthétiques de la part du réalisateur, qui ont fait qu'alors qu'on s'attendait au classique vert sur noir, l'on a tout un univers en bleu sur noir.

Soulignons cependant que le travail qui a été fait autour de la ligne de commande sur *Tron : Legacy* est assez particulier. En effet, comme il l'explique sur son blog, le designer en charge de réaliser les interfaces que l'on voit dans le film est allé puiser dans ce à quoi ressemblait une vraie ligne de commande aujourd'hui pour concevoir les interfaces du film : « *In addition to visual effects, I was asked to record myself using a unix terminal doing technologically feasible things.* ³⁶ », dit-il sur son blog ³⁷.

Ce designer a pris une esthétique et un imaginaire déjà existants et déjà très travaillés, basés eux-mêmes sur un imaginaire de la technologie qui a subi d'inévitables modifications suite à son adaptation au cinéma. Il a décidé de réinvestir cet imagi-

36. *En plus des effets visuels, l'on m'a demandé d'utiliser un terminal Unix qui faisait des choses techniquement faisables.*

37. NIMOY, Joshua T., « *Tron Legacy* » : [disponible en ligne : <http://jtnimoy.net/workviewer.php?q=178>], publié en 2010, consulté le 10 juin 2012



naire, cette esthétique, de leur insuffler de nouvelles choses –qui consistent en fait en un retour aux sources, et à cet imaginaire du code qui s’est dilué dans l’esthétique du premier film. Et ce qu’il insuffle au film, ce n’est pas l’imaginaire du code que l’on avait dans les années 1980, mais celui que l’on a en 2011 (et avec le bagage technologique qui est le sien), et qui n’est pas tout à fait la même chose. On a bien ici deux gestes de *remix* autour de la ligne de commande, à une échelle commerciale, pour le même film.

Troisième cas, tout aussi intéressant : le film *Matrix* a fait tout un travail esthétique sur le code et la ligne de commande en représentant la Matrice par du code écrit dans un langage mystérieux, défilant en vert sur fond noir, de haut en bas. On a là un rappel évident des vieilles consoles où le code s’affichait en vert sur un écran noir, rappel d’autant plus flagrant qu’il se trouve que ce code qui défile sur l’écran est le code source de la Matrice.

Cette image du code et du terminal est déjà, à ce stade, un remix de l’image classique que l’on a de cet objet. C’est déjà une manière de réactiver un réseau d’images et de discours autour de la ligne de commande, du code, voire du piratage informatique, sur lequel le film s’appuie. Ce seront alors non pas les images initiales de la ligne de commande (sans doute obsolètes), mais celles –déjà remixées, transformées, passées au filtre d’Hollywood– de *Matrix* qui seront réutilisées pour convoquer ce que signifie l’image de départ (l’expertise, une certaine frayeur. . .). Mais, ce faisant, l’image diffusée (celle de la Matrice) ajoute à la première image convoquée son propre réseau de discours et d’images, à savoir tout l’univers du film, sa philosophie, ses répliques cultes, le personnage de Neo. . .

C’est ce qu’il se passe lorsqu’on utilise un petit programme nommé « CMatrix³⁸ » qui permet de donner à ce qu’on est en train de taper dans son terminal l’apparence du code de la Matrice. Mieux : l’on peut aussi, en utilisant un petit script, générer soi-même une petite Matrice à l’intérieur de son terminal.

Nous voyons ici se dessiner une logique circulaire : nous avons un objet de départ –le *shell* Unix– qui se trouve investi par des images et des valeurs, images et valeurs que l’on voit détournées à des fins commerciales. Le produit de ce détournement et les imaginaires qu’il a généré vont se retrouver, à son tour, utilisé, réactivé, supplantant

38. « CMatrix » [disponible en ligne : <http://www.asty.org/cmatriX>], consulté le 31 août 2012



presque les imaginaires de départ : l'on va utiliser un terminal aux couleurs de *Tron*, à cause du film, sans se demander d'où viennent les choix esthétiques du film.

A ce stade de notre travail, nous y voyons maintenant plus clair dans la manière dont les imaginaires attendant à la ligne de commande circulent, porteurs de discours d'escorte, et sont réactivés. Ils vont influencer l'utilisation de la ligne de commande, de près ou de loin. Plongeons donc au milieu des usages, pour suivre ces imaginaires et voir mesurer leur 'influence.





Cette œuvre est mise à disposition sous licence Attribution - Pas d'Utilisation Commerciale - Partage dans les Mêmes Conditions 3.0 France. Pour voir une copie de cette licence, visitez <http://creativecommons.org/licenses/by-nc-sa/3.0/fr/> ou écrivez à Creative Commons, 44 Castro Street, Suite 900, Mountain View, California, 94041, USA.

2

Un usage entre rituel et braconnage

Pour Deleuze, « *la technologie est sociale avant d'être technique*¹ ». En effet, non seulement la technologie est le fruit du travail d'hommes et de femmes ancrés eux mêmes dans une société, mais elle fait également partie intégrante de nos vies. L'informatique est aujourd'hui un moyen de communication et un outil de travail très répandu. Elle participe donc au jeu social au même titre que les autres objets que nous utilisons quotidiennement. Par exemple, un ordinateur est aussi « social » qu'un téléphone ou qu'une voiture : nous utilisons ces outils quotidiennement. De plus, la pensée de leur concepteur est inscrite dans ces objets, et elle influence l'usage inévitable que nous allons en faire socialement. Du point de vue de la présentation de soi, par exemple, rouler en Mercedes ou en Twingo n'est pas la même chose. La technologie est donc d'abord sociale : la *manière* dont on use de la technique, le *sens* qu'on met dans cet usage sont déterminants.

Tous les discours et images qui gravitent autour de la ligne de commande, et que nous venons d'analyser, doivent donc être mis en regard avec la manière dont cet outil est utilisé.

2.1 Une logique de recommandation

Tout d'abord, regardons du côté de l'initiation à la ligne de commande. Comment en vient-on à utiliser cet outil ? Les imaginaires que nous avons précédemment analysés jouent-ils un rôle ?

1. DELEUZE, Gilles et GUATTARI, Félix, *Capitalisme et schizophrénie, tome 2 : Mille Plateaux*, Paris, Éditions de Minuit, 1980.



Les témoignages recueillis lors des entretiens² mettent en valeur une chose : la ligne de commande est un outil qui se transmet *via* une relation sociale que nous pourrions qualifier « de recommandation », où une personne « qui sait » va donner l'idée à une personne de son entourage d'utiliser la ligne de commande. Les deux scénarios principaux étant :

- les études : un professeur les a initiés
- les proches : ils ne sont pas étudiants en informatique ou informaticiens, mais un connaisseur leur a montré.

Il y a donc toujours une valeur de confiance qui s'installe : l'on croit celui qui nous montre l'outil. Inutile de dire que les discours d'escorte que nous avons vus se répandent ainsi très facilement, puisqu'un utilisateur qui recommande un outil est *a minima* satisfait, voire tout à fait convaincu.

Par exemple, deux des personnes que nous avons interrogées venaient de formation en informatique, où ils ont appris à utiliser la ligne de commande :

Je suis venu à l'utiliser à travers ma formation à l'école d'ingé. On nous a mis sur des terminaux. On a eu un TP "Utiliser la ligne de commande sous Linux". On passait notre temps sur des terminaux Sun Solaris tous pourraves. Pour avoir la même chose partout, y'avait que la ligne de commande pour arriver à ça. Et puis je me suis aperçu que je pouvais utiliser la ligne de commande pour me connecter en ssh à des machines plus puissantes.

Autre cas, qui combine les deux scénarios que nous avons exposés : lors d'un entretien, un utilisateur de la ligne de commande nous explique comment il a appris la ligne de commande de manière tout à fait autodidacte (motivé par un ami « *en gros une fois x m'a vanné, ça m'a énervé contre moi-même noob que j'étais, j'ai donc pris mon browser préféré et filé sur Wikipedia pour comprendre* »), il a appris la ligne de commande, qu'il a ensuite apprise à son père : « *je parle d'informatique avec mes amis et ma famille pour les convaincre et leur apprendre à sécuriser leur surf sur le net, protéger leur disque, leur routeur,...Là par exemple mon père se tire dans un pays craignos, je lui ai donc appris à utiliser Tails, à se connecter via un VPN et à chiffrer son disque.* »

Ce que l'on peut également remarquer, c'est que cette relation de recommandation qui sous-tend l'initiation à la ligne de commande va dans le sens d'un usage marginal

2. disponibles en annexe p.99



de l'informatique, restreint à un certain milieu (surtout quand il s'agit d'un apprentissage centré sur la sécurité informatique comme celui qu'a reçu le père de notre interlocuteur). Il faut au préalable connaître la personne qui va vous recommander l'outil pour en entendre parler. Il est de ce fait parfaitement logique que la ligne de commande soit principalement utilisée dans des milieux centrés sur l'informatique.

Dans un entretien, un utilisateur d'Ubuntu explique qu'il s'est progressivement mis à la ligne de commande grâce à cette distribution Linux : « *Quand j'ai commencé Ubuntu, je faisais pas du tout de ligne de commande et c'est arrivé un peu par accident, et j'ai fini par me dire, ah c'est mieux. [...] Le terminal est là, accessible, et y'a beaucoup de gens qui sont prêts à te montrer comment ça marche...* ». Ce qu'on remarque, c'est que l'adoption de l'outil se fait en trois temps : on lui montre d'abord l'outil, on le lui vante, puis il s'aperçoit qu'il y a du support disponible. Enfin, il l'adopte (« ah c'est mieux »). La valeur sociale est donc très importante dans l'adoption de l'outil, et ce n'est pas seulement une affaire de recommandation : il y a la certitude d'un lien social plus ou moins constant (« il y a beaucoup de gens prêts à te montrer comment ça marche »). Constat confirmé par un autre entretien : « *C'est une des particularités de l'informatique telle que je la conçois et à laquelle le logiciel libre participe. C'est un outil social comme les autres, et un outil de communication, pour favoriser la communication. Tout seul je ne ferai pas d'informatique.* »

Les imaginaires attendant à la ligne de commande ont leur rôle à jouer dans ce processus de recommandation. Lors d'un entretien, on nous raconte : « *Ça m'est jamais arrivé de mettre quelqu'un devant [la ligne de commande]. Ça m'est arrivé plus d'une fois d'animer des formations. Des gens qui venaient : "nous on veut apprendre à s'en servir", parce qu'on a remarqué que c'était quelque chose d'assez puissant.* » L'anecdote est éloquente : des personnes qui viennent à une formation en informatique pour apprendre à se servir de la ligne de commande parce qu'ils ont entendu que c'était un outil « puissant », ce qui est exactement le contenu des discours d'escorte que nous venons de mettre à jour.

Cette citation recueillie lors d'un entretien résume bien par quels « points d'accès » on arrive à connaître la ligne de commande : « *T'as besoin d'un élément déclencheur : soit la lecture de la documentation, soit l'observation de la pratique des autres, soit en direct, soit par Internet. "Tiens sa manière de faire me plaît je vais essayer."* Sachant que tu vas



passer à un moment ou un autre par la lecture de la documentation. » La dynamique sociale est effectivement importante, mais ni Internet ni la lecture de documentation par soi-même ne sont exclus : les imaginaires jouent aussi leur rôle. Ils soutiennent la logique de recommandation sociale, mais participent surtout à éveiller la curiosité des néophytes.

Un « milieu » de la ligne de commande ?

Sachant cela, nous serions tentés de penser que la ligne de commande est un outil dont l'usage est confiné à un milieu particulier, plutôt « geek », formé de passionnés d'informatique. Cette logique se tient : les verbatims renvoient que la plupart des usagers de la ligne de commande ont appris à s'en servir parce que quelqu'un le leur a conseillé, que ce soit un professeur (dans un cursus informatique ou scientifique) ou un ami (informaticien de formation ou autodidacte). Mais tous les usagers de la ligne de commande ne sont pas identifiables comme faisant partie d'un même milieu. Par exemple, l'un de nos interlocuteurs lors d'un entretien est étudiant à Sciences-Po, qui n'est pas réputée pour être une école d'informatique.

Nous sommes, en fait, assez loin d'une logique stricte de milieu : en ce qui concerne la distribution Linux Ubuntu, l'accent est très fortement mis sur la communauté, et sur une large communauté : des rencontres régulières ont lieu tous les mois à la Cité des Sciences, où se tiennent des conférences et des ateliers de vulgarisation. Il y a un forum, où les utilisateurs de la distribution s'entre-aident. De ce fait, les utilisateurs de cette distribution sont tous très différents : tous ne sont pas « geeks », beaucoup débutent, et l'on voit à quel point la présence de cette communauté surtout au début de l'apprentissage de la ligne de commande sous Linux, est importante. Il n'est pas rare d'entendre (ou de lire) de la part de ceux qui n'ont pas bénéficié de soutien : « j'ai arrêté, c'était trop compliqué ».

A des communautés rassemblées autour de distributions Linux, comme celle d'Ubuntu, s'ajoutent des lieux d'échanges dédiés à la ligne de commande sur Internet, comme la page Reddit consacrée à la ligne de commande, où l'on peut échanger conseils et astuces. Le fait qu'il y ait un lieu spécifique pour cela induit qu'il existe à minima un groupe d'utilisateurs plus ou moins défini. Mais cela suffit-il pour former un groupe social ? Ce serait comme considérer que les utilisateurs du système



d'exploitation Windows forment un groupe social, parce qu'ils partagent la même interface. Il nous semble qu'il y ait beaucoup trop peu de paramètres pour qu'on considère qu'il existe un « milieu » de la ligne de commande.

Nous aurions pu prendre pour base celle des différents utilisateurs de systèmes basés sur GNU/Linux, mais l'exemple d'Ubuntu prouve que l'on peut très bien être utilisateur d'un tel système sans toucher à la ligne de commande : parfois, le raccourci menant vers le terminal n'est pas accessible depuis le lanceur par défaut –c'est dire à quel point la base d'utilisateurs de la distribution est connue pour ne pas avoir expressément besoin de cet outil. Nous savons qu'il existe également des personnes qui ont appris à s'en servir seules, sans demander d'aide du tout, ce qui est le cas d'une des personnes que nous avons interrogées pour un entretien. Cela rend les contours d'un éventuel groupe social d'autant plus difficiles à tracer que l'activité sur les forums d'entraide de ces autodidactes étant nulle, nous ne pouvons estimer le nombre d'individus que cela représente. A cela s'ajoutent ceux qui ont appris à utiliser la ligne de commande au sein de leur famille, par exemple, et dont nous perdons également la trace. Non seulement il nous semble inexact, de parler de « groupe social », mais les contours de cet hypothétique groupe seraient difficiles à tracer.

Si l'on se rassemble autour de cet outil, s'il est le vecteur d'échanges et de sociabilité, il n'est donc, selon nous, pas adéquat de parler, en ce qui concerne la ligne de commande, de « milieu » ou de « groupe social ».

Un outil tout de même accessible

Si l'accès à la ligne de commande ne dépend pas exclusivement d'un lien social de recommandation, il est donc possible d'accéder à cet outil seul.

La communauté du logiciel libre permet la mise à disposition d'une large (et assez complète) documentation sur la ligne de commande sous GNU/Linux. Il est très facile de se procurer des documents PDF bien fournis, que l'on soit débutant, ou déjà un utilisateur avancé. Les documents que nous avons analysés en première partie ont été assez faciles à trouver.

S'ajoute à cela le fait que la ligne de commande ne vient pas seule : tout programme en ligne de commande sous GNU/Linux est livré avec sa page de manuel,

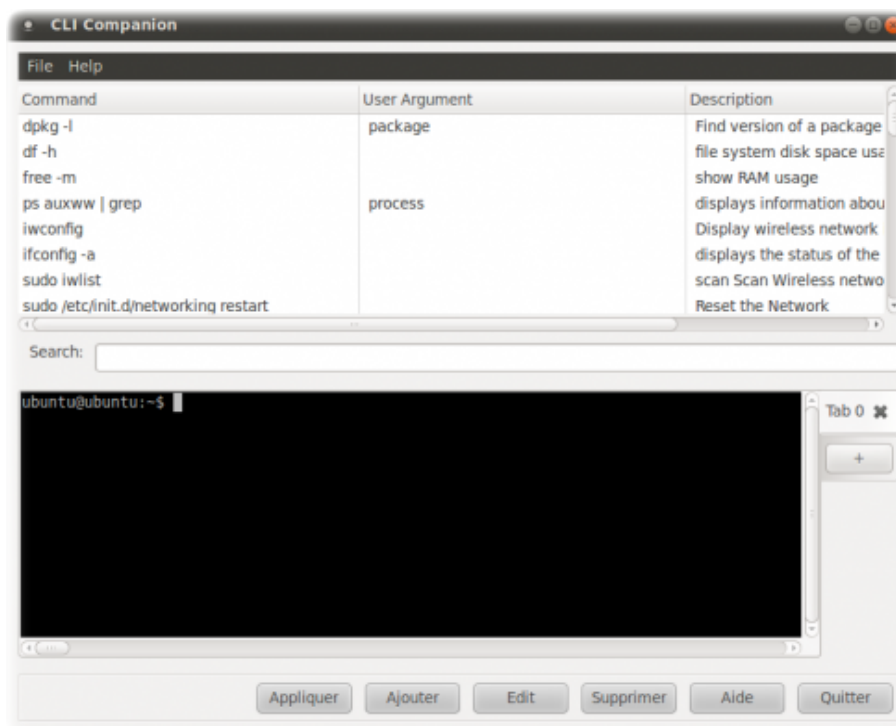


à laquelle on peut accéder via la commande *man* <nom du programme>. La « page de man » (traduction maladroite mais néanmoins très usitée de « *manpage* ») qui apparaît alors détaille toutes les options de la commande en question³.

Les pages de manuel servent régulièrement d'argument à ceux qui se voient rétorquer « je ne comprends pas comment utiliser la commande » par quelqu'un qui essaye d'apprendre quelque chose en ligne de commande : « lis d'abord le man » : « *En ligne de commande, tu lis le man, une fois que tu sais tu le fais.*, résume un de nos interlocuteurs lors d'un entretien.

Il existe même des programmes spécialisés pour apprendre la ligne de commande, comme *CLI Companion*, qui enveloppe le terminal d'une rassurante interface graphique, où l'on peut sélectionner les commandes désirées, soigneusement décrites dans le menu, et composer ainsi une suite de commandes correspondant à ses besoins en apprenant bien ce qu'elles font.

FIGURE 2.1: Interface de *CLI Companion*



Lors d'un entretien, une des personnes interrogées a noté que le « dé clic » pour aller vers la ligne de commande passait toujours par la lecture de documentation. C'est en quelque sorte un passage obligé.

3. Voir l'exemple de la page manuel de *man*, disponible en annexe, p.96



Cependant, même dans le cas d'un apprentissage autodidacte, la documentation ne suffit pas : on finit toujours par poser des questions à d'autres utilisateurs de la ligne de commande pour leur demander des conseils. La sociabilité est inévitable. C'est pour cela qu'Ubuntu met à disposition d'une part une documentation simple et accessible aux débutants, enrichie par des utilisateurs de la distribution, assortie d'un forum et d'un support très actifs. C'était nécessaire, car la lecture des pages de manuel peut paraître abrupte à un débutant.

Différentes courbes d'apprentissage : vers un élitisme de fait

De ce fait, un élitisme persiste. En effet, le facteur d'apprentissage est assez fort pour que, malgré l'accessibilité de la documentation et la disponibilité de personnes compétentes prêtes à donner des conseils, il reste que nombreux sont ceux qui abandonnent en cours de route.

Lors d'un entretien, un jeune autodidacte nous explique qu'apprendre seul la ligne de commande n'est pas si aisé que cela : si la documentation concernant les commandes est disponible, le fonctionnement global du *shell* est difficile à saisir. « *j'ai trouvé des lignes de commande facilement, mais des explications claires sur le fonctionnement global, non. J'ai dû fouiller, deviner certains trucs.* », dit-il.

Il n'est finalement pas aussi aisé que cela d'approcher la ligne de commande de but en blanc, sans soutien de la part d'autres utilisateurs de cet outil.

Sur ce point, l'analyse des cours⁴ de vulgarisation donnés sur la ligne de commande peut être intéressante.

Ce qui est dit dans l'un de ces documents, avec enthousiasme : « *Many people who give the command line a try are so amazed by its possibilities that they do not even want to go back to a graphical user interface!*⁵ », est donc peut-être ambitieux –et relève surtout du discours d'escorte. L'interface graphique apporte un confort auquel reviennent certains, découragés par l'apprentissage préalable qu'il faut faire pour utiliser la ligne de commande de manière efficace. En effet, la ligne de commande est un outil ingrat : les effets positifs de l'outil ne se voient pas immédiatement, contrairement à l'interface graphique où tout est fait pour être intuitif du premier coup.

4. dont nous avons un extrait en annexe, p.93

5. *Beaucoup de gens qui essayent la ligne de commande sont tellement impressionnés par ses possibilités qu'ils ne veulent même plus revenir à une interface graphique !*



La question à poser serait alors : « efficace, mais pour qui ? » : une sélection de fait s'opère entre ceux qui acceptent d'aller au bout de l'apprentissage et ceux qui abandonnent en cours de route.

Ainsi, la ligne de commande est un outil éminemment social. L'apprentissage de son utilisation, même s'il est possible seul, est en grande partie conditionné par des relations sociales. Mais cela va plus loin : l'utilisation même de cet outil peut être faite dans une optique de sociabilité.

2.2 Une logique de mise en scène de soi

La ligne de commande est clairement un outil qui signifie –on l'a vu–, un rapport différent à l'informatique, et l'on sait déjà que l'utilisation de l'outil est plutôt polarisée sur des milieux centrés sur l'informatique. L'hypothèse que nous pouvons faire maintenant est que la ligne de commande est utilisée non pas pour l'usage pour lequel il est prévu mais un usage *braconné* : se positionner en société comme ayant ce rapport différent à l'informatique.

Comment cet outil s'intègre dans une logique de mise en scène de soi

Nous avons vu comment les imaginaires étaient réactivés par leur utilisation dans un contexte social. Nous pouvons observer ce geste, qui consiste à lancer une commande uniquement pour faire peur à des gens de son entourage, comme un usage « braconné », détourné de la ligne de commande : celle-ci sert à utiliser un ordinateur, non pas à se positionner par rapport aux autres. Nous faisons en effet l'hypothèse que la ligne de commande fonctionne comme un outil de présentation de soi, et, à ce titre, peut servir de signe de reconnaissance pour une communauté. Une communauté qui, étant minoritaire, se construit une culture propre, selon ce qu'expose Howard Becker dans *Outsiders* :

« On a souvent suggéré qu'une culture apparaît chaque fois qu'un groupe de personnes se trouve confronté à un même problème, et dans la mesure où les membres de ce groupe sont capables d'entrer en interaction les uns avec les autres.



Ceux qui participent à des activités considérées comme déviantes rencontrent ainsi un problème typique : leur conception de ce qu'ils font n'est pas partagée par les autres membres de la société. »⁶

Etant donné qu'aujourd'hui, au niveau du marché des systèmes d'exploitation, GNU/Linux ne représente que quelques pourcents de la totalité des OS (systèmes d'exploitation) utilisés dans le monde, et que les autres OS sont connus pour être en interface graphique, la ligne de commande est, si nous nous en tenons à ces données très simples, l'affaire d'une minorité de personnes, visiblement les seules à pratiquer l'informatique de cette façon. Même si nous savons que tous les utilisateurs de GNU/Linux ne sont pas utilisateurs de ligne de commande, loin s'en faut (et c'est pour cela que nous avons émis des réserves quand à l'association de la ligne de commande à un « milieu » donné), il reste que cet usage est marginal, et qu'à ce titre, sans qu'il y ait forcément ni de communauté cohérente, l'on puisse se revendiquer de l'utilisation de cet outil.

Nous pouvons aller plus loin : dans le même ouvrage, Becker remarque, à propos des musiciens de jazz qu'il étudie, que ceux-ci ont une tendance certaine à ce qu'il appelle l'« autoségrégation ». En effet, ces derniers, par exemple, mettent des chaises pour se séparer du public, les « *caves* » (ceux qui ne comprennent pas le jazz) qu'ils méprisent et dont ils veulent éviter le contact, préférant rester entre eux.

Au cours de notre observation, nous avons découvert l'existence d'un forum en ligne de commande⁷, qui semble bien être le fruit d'une attitude d'autoségrégation.

FIGURE 2.2: Capture d'écran du forum U413.com

```
> Received "Login"
> Username?
> Received "PixelatedGeek"
> Password?

> |
```

Ce forum, pour commencer, n'est accessible que par invitation : il faut donc être « parrainé » pour obtenir un accès. C'est une protection déjà très grande du « monde

6. BECKER, Howard S., *Outsiders*, Métailié, 1985, p.104

7. Link, « U413.com - A command line forum », [disponible en ligne : <http://pixelatedgeek.com/2010/04/u413-com-a-command-line-forum/>], publié le 22 avril 2010, consulté le 21 août 2012



extérieur », il s'agit de rester de rester entre connaisseurs. Ensuite, l'interface du forum fonctionne vraiment comme un véritable *shell* Unix : si l'on ne sait pas véritablement s'en servir, on est relégué de fait, dans l'impossibilité de naviguer sur le site. Il y a donc un double filtre : le parrainage et le fonctionnement du site. Et c'est la ligne de commande, qui, ici, sert à la fois de filtre et de signe de ralliement.

De même (dans une moindre mesure) avec la page consacrée à la ligne de commande de Reddit⁸ : le forum arbore un fond noir, des onglets en vert, il y a la vache de la commande *cowsay*⁹ (figure 3.8) à droite. Ici, les références sont très importantes, elles ne sont compréhensibles que par des personnes qui connaissent un minimum la ligne de commande (et ont lancé au moins une fois *cowsay*), c'est là où réside le filtrage.

Il se passe également le même phénomène avec le site « *commandlinefu* »¹⁰, qui reprend les mêmes codes, les mêmes couleurs.

On notera par ailleurs que les verbatims présentés en première partie sont tous des discours mettant en scène cette différence : en se mettant en scène comme étant « incompris » par son entourage, l'interrogé se place en situation de marginalité, mettant le doigt sur une stigmatisation qui n'est peut-être pas voulue comme telle par son entourage, en utilisant comme matériau les imaginaires que nous avons analysés.

Ainsi, la ligne de commande est plus qu'un outil informatique : on peut en faire usage en société, pour se positionner comme différent par rapport à une majorité de gens ayant adopté des interfaces graphiques, revendiquant quelque part par là un point de vue différent sur l'informatique.

Un outil configurable, qui permet une certaine expression de sa personnalité

Dans la page de Reddit consacrée à la ligne de commande, que nous avons suivie pendant plusieurs mois, l'on voit que parmi les informations les plus fréquemment considérées comme intéressantes, nombreuses étaient celles qui concernaient la configuration, la personnalisation de son interface en ligne de commande. Comment

8. <http://www.reddit.com/r/commandline>, consulté le 31 août 2012

9. Commande dont le seul but est de produire une vache en ascii art, assortie d'un message

10. <http://www.commandlinefu.com>, consulté le 31 août 2012



rajouter un fond d'écran à son émulateur de terminal, comment donner à son terminal une présentation comme dans le film *Matrix*, mais aussi comment configurer ses raccourcis clavier de telle ou telle manière. La pratique du partage des fichiers de configuration est très répandue.

On en déduit assez facilement l'injonction sociale qu'il peut y avoir en surplomb à personnaliser son terminal. Les personnes qui s'en tiennent à la configuration par défaut du programme sont souvent montrées du doigt.

Lors d'un entretien, l'on nous explique cela :

« La ligne de commande n'est pas unique. Tu as différentes interfaces à la ligne de commande et différents interpréteurs. Le premier choix s'opère là. Tu peux décider de configurer d'une manière ou d'une autre. Déjà tu as un premier stade de configuration. Ensuite tu as pléthore de configurations avec une partie que tu vas repomper, sur des copains, sur Internet. "Tiens je ne connaissais pas cette commande, elle fait quoi ?" [...] Y'a des choses où tu vas procéder par essai. Y'a une partie par mimétisme, y'a une partie par documentation, y'a une partie d'habitude. C'est vraiment configurer ta ligne de commande pour qu'elle réponde à tes besoins, à tes envies. Mon fichier de configuration de zsh, mon shell, fait plus de 200 lignes. »

L'on remarque que ce geste de personnalisation est, lui aussi, lié à de la sociabilité : on s'échange des conseils et des captures d'écran, cela se fait au fil des rencontres et de l'utilisation de l'outil. En plus de cela, la personnalisation du *shell* permet d'adapter l'outil à ses besoins, mais aussi de le faire refléter sa personnalité.

Une logique qui va jusqu'à la théâtralisation

Cette logique de présentation de soi liée à la ligne de commande peut aller encore plus loin.

Nous avons pu remarquer précédemment que les imaginaires attendant à la ligne de commande n'étaient pas subis par les personnes concernées : au contraire, elles en jouent. Le meilleur exemple de l'attitude de ces utilisateurs de la ligne de commande, qui savent très bien ce qu'on dit d'eux quand ils ont fait l'usage est la petite application en ligne *Hacker Typer*¹¹. Sur cette page web, on peut trouver un petit terminal

11. « Hacker Typer », [disponible en ligne : <http://www.hackertyper.net>], consulté le 31 août 2012



en fond noir, où la police est de l'habituelle couleur verte, et où, quoi que l'on tape au clavier, il s'affiche à l'écran des commandes qui pourraient paraître complexes à une tierce personne qui nous regarderait faire. Comme le terminal est assez petit, l'on voit bien que l'on est pas vraiment sur le terminal de l'ordinateur. C'est parfaitement assumé comme étant faux.

*CMatrix*¹², le petit plug-in qui permet de faire ressembler votre émulateur de terminal à celui de Matrix, joue sur un ressort un peu différent. Il reprend un imaginaire déjà transformé de la ligne de commande (avec les signes cryptiques et très stylisés qu'on voit dans le film) pour le réinjecter dans le quotidien de l'utilisation d'un terminal. Cela revient à reprendre toute cette imagerie à son compte.

Autre aspect de cet côté théâtralisé de l'utilisation du terminal à des fins de présentation de soi : le forum uniquement en ligne de commande *U413.com*. On n'est pas ici dans la présentation de soi (par rapport à d'autres qui seraient différents), mais plutôt dans la prise de conscience et la mise en avant que l'on appartient à un groupe particulier.

Nous venons de voir comment la ligne de commande, chargée des imaginaires que l'on sait, peut être utilisée à des fins de positionnement en société. Mais *quid* des évolutions d'un tel usage ?

2.3 Un « rituel d'usage »

Nous proposons d'appeler « rituel d'usage » la manière dont l'usage de la ligne de commande évolue et se transmet : cela va de la simple reproduction du rituel de personnalisation à une attitude presque doctrinale.

Attardons-nous un instant sur la formulation que nous venons de proposer. Nous avons choisi *rituel*, dont la définition est –selon le *Larousse*– « Ensemble des règles et des habitudes fixées par la tradition », mais aussi « Ensemble d'actes, de paroles et d'objets, codifiés de façon stricte, fondé sur la croyance en l'efficacité d'entité non-humaines et approprié à des situations spécifiques de l'existence. ». La connotation religieuse du mot nous semble tout à fait pertinente ici. En effet, nous sommes en présence d'un outil informatique (donc bien « non humain ») en l'efficacité duquel on croit. Choisir la ligne de

12. « CMatrix » [disponible en ligne : <http://www.asty.org/cmatrix>], consulté le 31 août 2012



commande se rapproche du choix religieux, surtout au début, lorsque rien ne prouve l'efficacité tant vantée par ailleurs de l'outil.

Nous avons ajouté « usage » pour restreindre la ritualité au simple fait d'utiliser un outil : il ne s'agit pas d'un rite aussi construit que celui d'une religion à part entière ou d'une tradition, mais d'un usage répété et investi de la croyance en l'efficacité de l'outil.

Ce qui ressort des entretiens, c'est cette valeur d'habitude : l'apprentissage plus ou moins long et plus ou moins difficile de la ligne de commande a porté ses fruits et ses utilisateurs finissent par transposer les réflexes d'usage qu'ils ont acquis sur les ordinateurs des autres par exemple. Rappelons également que dans le cas où cet outil a été approché pendant les études, où il est plus facile de s'habituer à des outils qu'on utilise pendant tout le reste de sa carrière.

C'est cela que nous désignons par « rituel d'usage » : cette capacité à adopter des usages de manière quasi-aveugle (puisque rien ne nous assure *a priori* que nous allons en tirer bénéfice) que l'on va non seulement reproduire sans beaucoup les changer au cours de sa vie, et chercher parfois à transmettre aux autres quand l'occasion se présente.

Une « idéologie » de la ligne de commande

Les utilisateurs de la ligne de commande, on l'a vu, sont assez prompts à défendre leur outil. Ce sont eux les principaux porteurs des discours d'escorte que nous avons étudiés. S'il nous semble abusif de rattacher la ligne de commande à un milieu ou un groupe social donné, étant donné que les contours de celui-ci seraient beaucoup trop flous pour être intéressants, il nous semble cependant que la ligne de commande est un outil particulier en informatique parce que ses utilisateurs ressentent le besoin de justifier et de défendre leur choix en permanence.

Becker explique ce phénomène dans son ouvrage *Outsiders*¹³ :

La plupart des groupes déviants ont un système d'autojustification (une « idéologie »), même s'il est rarement aussi élaboré que celui des homosexuels. De tels systèmes contribuent certes, comme on l'a déjà noté, à neutraliser les restes

13. BECKER, Howard S., *Outsiders*, Métailié, 1985



d'attitudes conformistes que les déviants peuvent éprouver à l'égard de leur propre comportement ; mais ils remplissent encore une autre fonction : ils fournissent à l'individu des raisons solides, à ses yeux, de maintenir la ligne de conduite dans laquelle il s'est engagé.

Entendons-nous bien : le mot « déviant » dans le vocabulaire de Becker n'est chargé d'aucune connotation péjorative, il désigne les groupes minoritaires, discriminés par le reste de la population, marginalisés comme le seraient des hors-la-loi, puisqu'ils *dévient* du chemin tracé par la société. Cela concerne autant les joueurs de jazz que les homosexuels.

Avec les usagers de la ligne de commande, nous avons, selon Becker, un « groupe déviant » : il est évident que l'usage de la ligne de commande est minoritaire, et, étant donnés les imaginaires que nous avons étudiés, non pas discriminé mais clairement désigné comme marginal. Nous pouvons remarquer que la théorie de Becker, s'applique ici : se faire le porte-parole de discours d'escorte euphoriques est une manière de positionner socialement son choix comme bon, à ses yeux comme aux yeux des autres.

L'exemple de la page web recensant les raisons pour lesquelles la ligne de commande est un outil merveilleux est très parlant à ce sujet : l'auteur de l'article auquel Wired¹⁴ fait référence en a fait les frais : il met l'accent sur la virulence avec laquelle les utilisateurs de la CLI défendent leur outil lorsqu'on vante les GUIs : « *As he likely expected, the Linux-happy commenters at TuxRadar didn't take kindly to his "corrections". Dozens vehemently defended the command line, insisting it still has a very important place in the world of computing.*¹⁵ », dit l'article.

Nous avons bien ici une « idéologie » qui se dessine : les utilisateurs de la ligne de commande ont une tendance très prononcée à défendre leur outil, autant pour convaincre les autres que pour se convaincre eux-mêmes, parfois. Ce qui peut donner lieu à des comportements excessifs.

14. GARLING, Caleb, « Why the GUI Will Never Kill the Sacred Command Line », *Wired Enterprise*, [disponible en ligne : <http://www.wired.com/wiredenterprise/2012/07/command-line>], publié le 10 juillet 2012, consulté le 27 juillet 2012

15. *Comme il s'y est sûrement attendu, les commentateurs pro-Linux de TuxRadar n'ont pas très bien pris ses "corrections". Des dizaines ont défendu la ligne de commande avec véhémence, insistant que le fait qu'elle a une place très importante dans le monde de l'informatique.*



Des usages excessifs

Les utilisateurs de la ligne de commande sont à ce point dans un usage ritualisé que celui-ci finit par se rigidifier : les usages peuvent devenir excessifs par rapport à l'intérêt réel que l'on a à tirer de la ligne de commande.

Le meilleur exemple est certainement celui-ci, recueilli lors d'un entretien : « *Dès que t'as un truc un peu répétitif à faire, tu vas le faire en ligne de commande. Je suis même prêt à passer deux heures et demi à écrire un script pour le faire de façon élégante alors que ça prenait moins de temps de le faire à la main* ». Nous avons ici un usage vraiment contre-productif de la ligne de commande : il y passe beaucoup de temps, pour quelque chose qui n'en valait peut-être pas la peine.

Autre exemple : nous avons remarqué au Nicelab, le hackerspace que nous fréquentions, que certains sont tellement heureux d'utiliser leur outil qu'ils décident de ne pas utiliser (ou d'utiliser le moins possible) leur souris, en mettant par exemple du scotch en dessous, pour l'empêcher de fonctionner.

Ils passent également par des détours pour arriver à leurs fins quand il s'agit d'interagir avec des documents de type graphique : « *Tes usages, tes habitudes te déforment aussi. En ce qui me concerne, je ne vais pas regarder de vidéos sur Internet, ou assez peu d'images. Ou je vais télécharger l'image dans un fichier temporaire et l'afficher* »

Nous pourrions certainement trouver encore d'autres exemples. Le peu que nous avons à notre disposition nous montre bien comment un usage peut devenir rigide, voir absurde, en tous les cas, excessif.

Au final, nous pouvons affirmer que la ligne de commande est utilisée dans une optique identitaire, mais il serait tout à fait inexact de réduire cette identité à un milieu social précis. Ce que l'on revendique avec la ligne de commande est beaucoup plus proche d'une idée que l'on se fait de l'informatique que l'appartenance à un groupe social donné. Nous avons vu jusqu'où s'étendait l'influence des imaginaires étudiés en première partie : c'est eux qui s'incarnent dans l'utilisation de la ligne de commande en contexte social.

Mais cette sociabilité n'est pas neutre : toute relation sociale est empreinte de relations de pouvoir, d'interactions entre différentes instances disposant de plus ou moins de marge de manoeuvre, et nous voyons bien que si des usages sont modifiés,



des discours diffusés, ce n'est pas par hasard. Nous ne pouvons en rester là dans notre réflexion : il nous faut mener une réflexion plus politique sur notre objet, pour comprendre ce que sous-tendent ces usages.



3

Des rapports de force et des compromis

Latour considère¹ que la sociologie se pratique de manière plus efficace non pas en restant en surplomb et en tentant d'appliquer des modèles théoriques à ce que l'on constate, mais plutôt de chercher par quels réseaux le pouvoir circule autour de l'objet ou au sein du milieu qu'on étudie. C'est ce à quoi nous allons nous attacher ici.

En faisant référence au pouvoir, nous faisons écho au travail de Foucault, pour qui le pouvoir est d'abord « la capacité de faire certains types de mouvements »². Une « capacité », qui, possédée par des êtres humains, éminemment sociaux, prend fatalement une valeur sociale : « *Les sortes d'actions dont le corps humain est capable dépendent, d'une part, de sa constitution physique et morale, et, d'autre part, des relations sociales au sein desquelles il vit.* »³

Ce pouvoir-là s'exerce donc lorsqu'un corps humain, individuel ou collectif (une société, une institution) agit sur des actions d'autrui.

« De ce pouvoir il faut distinguer d'abord celui qu'on exerce sur les choses, et qui donne la capacité de les modifier, de les utiliser, de les consommer ou de les détruire - un pouvoir qui renvoie à des aptitudes directement inscrites dans le corps ou médiatisées par des relais instrumentaux. Disons qu'il s'agit là de « capacité » Ce qui caractérise en revanche le « pouvoir » qu'il s'agit d'analyser ici, c'est qu'il met en jeu des relations entre individus (ou entre groupes). Car il ne faut pas s'y

1. LATOUR, Bruno, *Reassembling the social : An Introduction to Actor-Network-Theory*, Oxford, Oxford University Press, 2005

2. PATTON Paul, *Le sujet de pouvoir chez Foucault*, in : *Sociologie et sociétés*, vol. 24, n° 1, 1992, p. 91-102.

3. FOUCAULT, Michel, *Dits et écrits, Tome IV*, Paris, Gallimard, 2001



tromper : si on parle du pouvoir des lois, des institutions ou des idéologies, si on parle de structures ou de mécanismes de pouvoir, c'est dans la mesure seulement où on suppose que « certains » exercent un pouvoir sur d'autres. Le terme de « pouvoir » désigne des relations entre « partenaires » (et par là je ne pense pas à un système de jeu, mais simplement, et en restant pour l'instant dans la plus grande généralité, à un ensemble d'actions qui s'induisent et se répondent les unes les autres).⁴ »

Nous entamons donc ici une réflexion moins sociale que politique, sur des outils dont on ne soupçonne pas la force politique de prime abord. Pourtant, les jeux de responsabilités croisées entre le développeur, l'utilisateur, et le contexte dans lequel il vit et utilise l'outil ont quelque chose du jeu de responsabilités qu'on retrouve entre des hommes politiques, des citoyens et une république. C'est en cela que l'enjeu de notre travail est communicationnel : il ne s'agit pas –bien sûr– de communication au sens strict du terme (à savoir, d'après le Larousse, d'« échange verbal entre un locuteur et un interlocuteur ») mais il y a bien ici un débat, une scène sur laquelle ont lieu des échanges, des « relations entre « partenaires » ». Il y a, au sens large, de la communication.

Ce que nous appelons « réseaux de pouvoir » fait référence à cet « ensemble d'actions qui s'induisent et se répondent les unes les autres » : nous avons différentes instances (le développeur, les utilisateurs finaux, l'entourage de ces derniers, les constructeurs informatiques, ...), qui entretiennent entre elles des liens faits d'actions exercées les uns sur les autres. Pour bien comprendre comment s'exercent les jeux d'influence que nous avons vu se dessiner dans la partie précédente, cartographions, en quelque sorte, ces réseaux.

Cet exercice de mise au jour de relations de pouvoir nous semble d'autant plus intéressante que, d'après Foucault, « Là où il y a pouvoir, il y a résistance »⁵. C'est la *résistance* qui nous intéresse ici : davantage que les logiques de pouvoir dans lesquelles est pris l'utilisateur, nous cherchons à mesurer la marge de manoeuvre qu'il peut se ménager vis-à-vis d'elles.

4. FOUCAULT, Michel, *Ibid.*

5. FOUCAULT, Michel, *Histoire de la sexualité I : La volonté de savoir*, Paris, Gallimard, p.125



3.1 Discipline du corps : une interface dont le pouvoir s'étend jusqu'à notre posture

Nous avons vu, au cours de l'étape précédente de notre réflexion, comment une interface, à savoir une certaine manière d'organiser des *signes* sur un *écran*, pouvait générer de la sociabilité dans le « monde réel » (en créant des situations d'étonnement, de peur, en poussant les utilisateurs à échanger à propos de leur outil). Selon nous, le pouvoir dont dispose la ligne de commande va plus loin que l'organisation d'un groupe d'utilisateurs autour d'elle : dans la droite ligne de la pensée de Foucault, nous pensons qu'elle exerce sur le corps de ses utilisateurs une certaine pression, leur impose une discipline.

Le clavier, la souris et les interfaces

A la base de cette réflexion, il y a le constat que les interfaces se sont développées, depuis le début des recherches sur les interfaces graphiques, de concert avec les dispositifs de pointage et d'entrée (la souris et le clavier).

Le clavier a été le premier dispositif que l'on a accolé à l'ordinateur : il s'est agit, dès les débuts de l'informatique, de manipuler du texte, et uniquement du texte. L'on a donc simplement importé le clavier, qui existait déjà sur les machines à écrire, pour l'adapter à l'ordinateur (avec la même disposition, donc que les machines à écrire de l'époque, pour ne forcer personne à réapprendre à taper à la machine⁶).

Le clavier tel que nous le connaissons n'a pas toujours eu cette forme. il est le fruit de la controverse de 1962 entre Douglas Engelhardt et Harold Wooster⁷. Engelhardt, par ailleurs concepteur de la souris, avait imaginé de la coupler avec un clavier non pas comme le nôtre –le clavier AZERTY en Europe, QWERTY outre-Atlantique– mais avec accord (nommé par son concepteur le CK). L'idée était de faire des combinaisons de touches pour écrire au lieu de taper les lettres une par une. L'idée d'Engelhardt était d'avoir un clavier en étroite collaboration avec la souris, afin que les gestes soient plus fluides.

6. BARDINI, Thierry, *Le clavier avec ou sans accord : retour sur une controverse oubliée*, in : *Réseaux*, 1998, volume 16 n° 87, pp. 59-62

7. BARDINI, Thierry, *Ibid.*, pp. 45-74



Mais son interlocuteur, ainsi que les mécènes du Xerox PARC ne l'entendaient pas de cette oreille. Ils considéraient en effet que l'invention d'Engelbart demandait un apprentissage trop important aux utilisateurs inexpérimentés pour être commercialisé à grande échelle. Ce sera finalement le clavier QWERTY qui sera intégré au système NLS (oN-Line System), le premier à permettre l'usage des liens hypertextes, de la souris, des fenêtres, des principaux concepts de l'informatique moderne.

Nous mettons l'accent sur le fait que cela a été pensé comme un *système* intégré : la souris a été pensée pour permettre la sélection à la volée de texte, l'usage des fenêtres. Les relations entre tous les éléments ont été prévues. C'est tout le sens du concept d'interface de type « WIMP » (Window, Icons, Menus, Pointer) qu'ont mis au point les chercheurs du Xerox PARC : la souris et le concept de « bureau » (*desktop*) ont été développés ensemble :

« When Doug Engelbart invented the mouse, he arrived at the dominant design for input devices in a single leap from the light pen, and the development of the mouse since has been more in the nature of evolution than revolution. Engelbart also invented the point-and-click text editor for the NLS system (oNLine System) that he developed at the Stanford Research Institute (SRI), and that system migrated with members of his design team to the fledgling Xerox PARC and became the foundation of the Alto, the first computer with a GUI. »⁸

L'usage de la ligne de commande laisse aujourd'hui une place hégémonique au clavier (à plus forte raison avec l'usage d'un gestionnaire de fenêtres fonctionnant quasi exclusivement au clavier, comme *Awesome*), qui est le plus ancien dispositif d'entrée dont l'informatique a été dotée si l'on ne considère pas les cartes perforées –qu'on utilisait avant le clavier pour entrer des données dans un ordinateur– comme tel. Ce dispositif, depuis l'invention de la machine à écrire dans les années 1870, n'a pas fondamentalement changé, bien que de nombreuses dispositions alternatives au QWERTY aient été trouvées depuis. Nous comprenons bien pourquoi le clavier est si important : d'une part il s'agit de faire usage essentiellement de texte, d'autre part, la

8. MOGGRIDGE, Bill, *Designing Interactions*, Cambridge, The MIT Press, 2006 (*Quand Doug Engelbart a inventé la souris, il est arrivé au design dominant pour les dispositifs de pointage en un seul saut à partir du crayon optique, et le développement de la souris depuis a été plus une évolution qu'une révolution. Engelbart a aussi inventé l'éditeur de texte pointer-cliquer du système NLS (oNLine System) qu'il a développé au Stanford Research Institute (SRI), et ce système est passé des membres de son équipe de design au Xerox PARC naissant et est devenu le fondement du premier ordinateur avec une interface graphique.*



souris est un dispositif de pointage très étroitement lié aux interfaces graphiques, que certains usagers de la ligne de commande rejettent.

Nous voyons bien comment interface et dispositifs de pointage et d'entrée sont liés, et comment le choix d'interface va déterminer les gestes et la posture que nous aurons devant l'écran. En effet, une interface dédiée à la ligne de commande est conçue pour que l'utilisateur aie en permanence les deux mains sur le clavier, alors que les interfaces graphiques font entrer en jeu la souris, responsable d'un déport régulier de la main droite vers elle, afin de pointer. L'attitude physique face à l'interface n'est pas la même selon le type d'interface utilisé. C'est pour cette raison que les GUI ont souvent reçu le sobriquet de « cliquodromes »⁹.

L'usage d'une interface n'est donc pas seulement un choix esthétique (au sens premier de « esthétique ») : c'est le choix d'une certaine attitude par rapport à l'ordinateur, c'est ancré dans le corps de l'utilisateur.

Vim : un logiciel en ligne de commande qui a laissé sa marque

Allons plus loin dans cette mise au jour d'une discipline du corps : il y a l'attitude générale que nous avons face à l'écran, mais il y a aussi des choses plus fines qui influencent nos gestes : les raccourcis clavier par exemple. En effet, lorsqu'on a appris des raccourcis clavier, on est tenté de les reproduire partout : c'est pour cette raison que la grande majorité des programmes sous Windows, par exemple, associent l'action d'annuler au raccourci *ctrl* + *z*.

Dans une interface en ligne de commande, les habitudes que nous prenons vis à vis de certains raccourcis clavier sont d'autant plus fortes et d'autant plus visibles qu'il n'y a pas l'usage de la souris pour contrebalancer celui du clavier, comme c'est le cas dans les interfaces graphiques.

Pour illustrer ce point, penchons-nous sur un programme en particulier : *Vim*. Très répandu, cet éditeur de texte en mode console a laissé son empreinte dans des usages de l'informatique bien éloignés de celui de l'écriture de lignes de code. Par exemple, il existe un plugin du navigateur web Firefox nommé *Vimperator*, qui permet d'avoir les mêmes raccourcis clavier que dans *Vim* dans le navigateur web *Firefox*.

9. Attitude résumée dans une déclaration de Eben Morglen par : « *Tu pointes et tu grognes* »



La présence de ce plugin dit une chose : c'est que l'utilisation quotidienne (ou du moins très très fréquente, dans le cas d'une personne dont le métier est d'écrire du code) de *Vim* a donné à tellement de personnes des habitudes assez fortes (celle de faire `:q` pour quitter ou `/` pour rechercher, par exemple), qu'il y a eu besoin de faire un plugin. Habitudes renforcées par le fait que ces commandes sont reprises dans d'autres applications en ligne de commande (il est possible de faire `:q` dans le client mail *Mutt*, également).

Le logiciel *Vim* est tellement répandu qu'il existe des méthodes pour influencer le fonctionnement de *bash*¹⁰, l'un des principaux langages utilisés pour faire de la ligne de commande, pour qu'il reconnaisse tous les raccourcis clavier valides dans *Vim*. On voit ici le programme se constituer comme une instance de pouvoir à part entière, à qui l'utilisateur délègue le choix de ses raccourcis clavier pour l'ensemble de ses activités en ligne de commande. Il accepte de répéter les mêmes gestes, déterminés par ce programme, dans des circonstances qui excèdent ce dernier.

L'influence de *Vim* sur la posture que nous adoptons face à la machine va beaucoup plus loin : les utilisateurs de la ligne de commande, qui souvent méprisent la souris et les GUI, sont ceux qui ont eu l'idée d'inventer leur propre interface matérielle avec le logiciel : une pédale¹¹ qui permet de passer en appuyant dessus d'un mode (*NORMAL* ou *INSERT*) à l'autre de *Vim*. On peut s'interroger sur l'avis qu'aurait eu Engelhardt à propos de cet objet.

Quoi qu'il en soit, nous avons ici un très bel exemple de la manière dont le pouvoir d'une interface (celle du programme *Vim*) va modifier la manière dont on dispose son corps autour de la machine, en mettant en place un geste supplémentaire, sur un dispositif supplémentaire.

10. DANEN, Vincent « Using vi bindings in bash and zsh », *TechRepublic* [disponible en ligne : <http://www.techrepublic.com/blog/opensource/using-vi-key-bindings-in-bash-and-zsh/193>], publié le 14 avril 2008, consulté le 31 août 2012

11. ROUGET Paul, « Une pédale d'embrayage pour *vim* », *Sheevaboite*, [disponible en ligne : <http://www.sheevaboite.fr/articles/pedale-embrayage-vim>], publié le 21 juin 2012, consulté le 31 août 2012



Un compromis de fait, décidé par les constructeurs informatiques

Mais ce pouvoir que l'interface fait peser sur nos gestes et notre posture n'est pas sans limites. En effet, la manière dont les ordinateurs sont aujourd'hui construits et assemblés impose à tous la présence d'un dispositif de pointage.

Il se trouve en effet qu'aujourd'hui tous les ordinateurs portables sortent d'usine avec un dispositif de pointage. De ce fait, les utilisateurs se voient forcés de composer avec la présence de ce dernier. C'est un compromis de fait : aujourd'hui, l'on n'imagine plus d'ordinateur sans souris. Cela a certainement influencé le fait que les principales distributions GNU/Linux proposent aujourd'hui des interfaces graphiques (certes, beaucoup plus légères que celles auxquelles Windows et Mac OS nous ont habitués : *Awesome*, par exemple) qui prennent en compte le support de la souris. Le choix qui est fait par les constructeurs est également dû au fait que les principaux systèmes d'exploitation aujourd'hui comportent des interfaces graphiques, pour lesquelles la souris est nécessaire. Comme la plupart des ordinateurs sont vendus avec un système d'exploitation préinstallé, les constructeurs s'alignent avec le système d'exploitation (à savoir pour une très large majorité des machines, Windows) en question.

Les machines n'arrivent pas neutres sur le marché : elles sont assemblées dans l'optique d'une utilisation quotidienne de la souris. Et ce choix fait de la part des constructeurs est le résultats d'accords économiques passées avec Microsoft, par exemple. Ce sont des logiques économiques qui dépassent de loin l'utilisation d'une interface, mais finissent par l'impacter : l'utilisateur n'a qu'une très faible marge de manoeuvre vis à vis de cela, il peut seulement choisir ou de ne pas utiliser la souris du tout, ou d'acter que la souris est parfois utile et en faire un usage ponctuel, puisque son système d'exploitation le lui permet.

3.2 Responsabilité et compromis

Nous en venons donc à ce qui incombe à l'utilisateur, dans cette cartographie des réseaux de pouvoir que nous sommes en train de tracer. Pour aborder ce point, nous souhaitons lier le concept foucaltien de pouvoir avec celui de *délégation* chez Rieder. Pouvoir et responsabilité sont en effet liés : la capacité de pouvoir faire certaines actions sous-entend le fait que nous devons assumer les conséquences de cette dernière.



Dans les relations de pouvoir autour de la ligne de commande que nous allons étudier, il y a donc la question de ce que nous déléguons à la machine comme responsabilité.

Cette réflexion trouve son fondement dans le travail de Latour :

« Nous admettons fort bien que les techniques soient le prolongement de nos organes. Nous savions qu'elles étaient la démultiplication de la force. Nous avons simplement oublié qu'elles étaient aussi la délégation de notre morale. »¹²

En effet, ce que les machines font à notre place reste quelque chose que nous faisons : nous ne faisons que *déléguer* au système la responsabilité de l'avoir fait à notre place. C'est ce que met en lumière cette citation : « Nous avons été capables de transférer à des non-humains non seulement la force, mais aussi des valeurs, des devoirs et une éthique »¹³.

Attachons-nous donc à chercher quelles sont les lignes de tension qui se tracent autour de l'usage de la ligne de commande, par quoi il est restreint.

Un élitisme de fait : une logique économique où le logiciel libre est marginal

La première source de normativité est économique : étant donné que la ligne de commande est principalement utilisée sous GNU/Linux, nous ne pouvons considérer la ligne de commande sans la placer dans l'écosystème plus large du marché des systèmes d'exploitation.

Replaçons tout cela dans son contexte historique et économique. Dans son essai *In the beginning was the command line*, Neil Stephenson raconte comment les premiers « ordinateurs personnels » sont arrivés sur le marché. : « *The Macintosh OS was a revolution in both the good and bad senses of that word. Obviously it was true that command line interfaces were not for everyone, and that it would be a good thing to make computers more accessible to a less technical audience—if not for altruistic reasons, then because those sorts of people constituted an incomparably vaster market.* »¹⁴

12. LATOUR, Bruno, *La clef de Berlin et autres leçons d'un amateur de sciences*, Paris, La Découverte, 1993, p. 32

13. LATOUR, Bruno, *Ibid.*, p. 62

14. STEPHENSON, Neal, *In the beginning was the command line*, 1999 (Macintosh OS a été une révolution dans le bon et le mauvais sens du mot. De toute évidence, il était vrai que les interfaces en ligne de commande n'étaient pas pour tout le monde, et qu'il était une bonne chose de fabriquer des ordinateurs plus accessibles à un public moins technophile –mais pas pour des raisons altruistes, mais parce que ces gens constituaient un marché incomparablement plus vaste.)



Si l'on étudie les estimations¹⁵ qui sont à notre disposition avec les précautions qui s'imposent, c'est à dire en sachant que nous n'avons aucune garantie sur ces chiffres puisque ce ne sont que des estimations (qui sont faites selon une méthodologie restant inconnue du public), basées sur l'achat des licences d'OS, mais ne tenant compte ni des copies d'OS piratées, ni des réinstallations qui ont lieu sur des ordinateurs dont l'OS a été préinstallé auparavant (nous sommes ici sur des chiffres concernant uniquement des ordinateurs de bureau), nous avons une idée du marché, au niveau des proportions, mais nous manquons de données précises et basées sur des méthodologies sûres pour avoir une idée précise de quels OS sont réellement utilisés.

Il reste que l'on remarque une tendance sur deux ans assez stable, avec une écrasante majorité des ordinateurs sur le marché sous Windows (toutes versions confondues), le reste sous Mac OS, et les différentes distributions Linux se partageant les pourcents restants.

Cette situation économique apporte un élément d'explication que nous n'avions pas à propos de l'imaginaire de l'expert lié à la ligne de commande : au delà des discours d'escorte, le marché des systèmes d'exploitation confine les utilisateurs de Linux (où la ligne de commande reste beaucoup plus présente que dans les autres) dans une situation de minorité, et donc de minorité « savante », puisqu'ils sont identifiés comme étant les seuls à encore utiliser de manière plus ou moins répétée, la ligne de commande.

On comprend facilement l'équilibre auquel on est arrivé aujourd'hui quand on sait que les systèmes d'exploitation ont été conçus pour rendre l'informatique accessible au grand public. Un grand public qui représente nécessairement un plus grand nombre de personnes que la somme de celles qui utilisent l'informatique à un niveau plus avancé.

Le logiciel libre : avoir le contrôle c'est une question de liberté

Le premier *sujet de pouvoir* que nous pouvons déceler en ce qui concerne la ligne de commande sont les développeurs, ceux que Lessig appelle les « auteurs de code » :

15. SANYAS, Nil, « OS de bureau : le marché n'évolue quasiment pas », *PC INpact*, [disponible en ligne : <http://www.pcinpact.com/news/65525-os-bureau-windows-mac-os-linux.html>], publié le, consulté le 7 août 2012 – basé sur les chiffres de *NetMarketShare* : <http://www.netmarketshare.com/operating-system-market-share.aspx?qprid=8&qpcustomd=0>, consultés le 7 août 2012



« *Tel que le monde est aujourd'hui, les auteurs de code deviennent, de manière croissante, des faiseurs de lois* »¹⁶

Richard Stallman, dans de nombreuses conférences, dont celle qu'il a donné à Sciences Po Paris en 2011¹⁷ explique le lien entre libertés individuelles et logiciel libre – la philosophie à l'origine de GNU/Linux. Il démontre que si l'on ne sait pas ce que fait sa machine, c'est que l'on ne la contrôle pas totalement, et donc que l'on n'est pas libre. Or, l'approche de l'informatique favorisée par la ligne de commande est par excellence une approche centrée sur le contrôle : la plupart des programmes en ligne de commande sont configurables uniquement *via* des fichiers de configuration qu'il faut éditer à la main, la plupart des programmes ont également des options « verbeuses », qui rendent compte à l'utilisateur de ce que le programme fait.

De plus, la liberté va de pair avec la responsabilité. Utiliser un GUI est certes plus pratique, mais l'on défasse du pouvoir aux agents qui servent d'intermédiaire entre nous et le matériel de l'ordinateur. A l'inverse, utiliser la ligne de commande, c'est assumer la responsabilité de son système : si une commande est erronée et que tout casse, l'utilisateur ne peut s'en prendre qu'à lui-même.

C'est ce que sous-tend le concept de *délégation* de Bernhard Rieder : il rapproche, dans son texte, l'informatique de la politique en expliquant que toutes les tâches confiées à des *agents* qui travaillent seuls sans intervention humaine sont autant de morceaux de pouvoir délégués à la machine. Rieder emprunte ce concept à la politique, et plus précisément à la pratique de la démocratie. En effet, la démocratie consiste en une opération de *délégation* de pouvoir du peuple à son ou ses représentants. Si l'on a accepté de déléguer une tâche à un agent, l'on défasse du pouvoir, mais l'on se dégage de cette tâche. Si on décide de l'effectuer pour garder le contrôle sur ce qui est fait, il faut en assumer les conséquences.

C'est le poids de cette responsabilité, que Stallman illustre parfaitement avec la rigidité de son exigence vis à vis du logiciel libre, qui amène les utilisateurs à défasser du pouvoir pour se délester.

16. LESSIG Lawrence, *Code and other Laws of Cyberspace*, New York, Basic Books, 1999, p.60

17. STALLMAN, Richard, « Une société des réseaux libéré », organisé par Libertés Numériques, l'association numériques à Sciences Po, 19 octobre 2011, amphithéâtre de Sciences Po Paris



Des compromis où on défausse, délègue du pouvoir

Malgré la défense véhémement qu'ils en font en public, les utilisateurs de la ligne de commande sont forcés de faire, dans leur quotidien, des compromis : il revient dans les entretiens une chose, c'est qu'une utilisation au quotidien de la ligne de commande sans toucher à une interface graphique est quasi impossible. Quelle marge de compromis se ménagent-ils ?

Rappelons qu'aujourd'hui très peu de personnes aujourd'hui utilisent –à l'exception des machines utilisées pour l'hébergement, entre autres, qui sont essentiellement administrées en ligne de commande– une interface entièrement dédiée à la ligne de commande. Comme nous l'avons vu dans notre avant-propos, la plupart des systèmes aujourd'hui incluent une « couche » graphique. On en conclut que très peu de personnes font une utilisation exclusive du *shell*.

On se rend compte en effet que l'on est forcé de défausser du pouvoir vis à vis de sa machine si l'on veut arriver à ses fins. Pour commencer, l'existence du Web rend inévitable l'utilisation d'une interface graphique. En effet, sauf à accepter de ne faire qu'une navigation en mode texte (et dans ce cas le navigateur *links*, qui permet cela à l'intérieur d'un terminal, suffit), l'on est obligé d'ouvrir un navigateur Web, si l'on veut profiter des vidéos ou des animations *javascript*.

Dans le même ordre d'idées, la présence de la souris est aussi source de compromis. L'éditeur de texte *Vim*, par exemple, ne propose pas la souris par défaut puisqu'il est censé n'être utilisé que *via* le clavier, mais l'on peut faire le choix d'activer l'option du support de la souris.

L'utilisateur est de fait limité par son outil, et doit choisir de quel côté faire un compromis, mais il n'est pas totalement libre de faire ce qu'il veut, en réalité. Ou il accepte de ne plus utiliser la souris, et dans ce cas de ne plus utiliser un seul programme nécessitant l'utilisation d'un pointeur, ou il choisit d'utiliser la souris, mais de faire un compromis sur son interface. Cela se vérifie dans les entretiens : « *En fait ce qu'il faut voir, c'est que la ligne de commande n'est pas adaptée à tous les usages. Par exemple, pour le Web, l'image et la vidéo. Or, ces usages et ces formes de communication, c'est une grosse grosse partie de la communication.* »

Il y a un discours d'escorte de la ligne de commande dont on a vu que les utilisateurs (au moins en grande partie) sont des porteurs. L'on a vu avec quel enthousiasme



ils encourageaient l'utilisation de leur outil. Et pourtant, ce sont ces mêmes utilisateurs que l'on surprend à faire des compromis sur l'intransigeante pensée du tout-clavier et du tout-CLI. L'on s'excuse, comme pour rendre hommage à la CLI malgré l'écart que l'on fait : « je sais que c'est mal, mais j'ai trouvé un raccourci pour prendre en compte la souris dans *Vim* ».

On voit que l'utilisateur délègue peut être moins de pouvoir avec ce type d'interface, mais d'une part il est forcé de faire tout de même des compromis, et d'autre part, il est pris dans une logique de représentation qui le dépasse, puisqu'il délègue une partie de ce qu'il signifie socialement aux yeux des autres à sa machine.

3.3 Esthétique de l'efficacité

Alors, ce qu'on rechercherait en utilisant la ligne de commande, ce ne serait pas l'efficacité et la puissance tant vantée par ses utilisateurs, mais juste une manière de se représenter aux autres... L'efficacité ne serait-elle qu'une question d'apparence ?

Platon et la ligne de commande

A première vue, nous serions tentés de répondre « non » à cette question. En effet, la ligne de commande nous semble nourrie d'une pensée de l'informatique qui tend à donner à l'utilisateur le contrôle le plus étendu possible sur sa machine, en utilisant un langage et une esthétique dépouillées de ce qui serait superflu.

Aussi incroyable que cela puisse paraître, cette pensée est très proche de celle de Platon. Dans le *Phédon*, notamment, il expose sa théorie des *Idées* ou formes intelligibles. Celles-ci ne sont accessibles que par l'intellect. Intellect que l'on fait progresser vers les *Idées* par l'art du dialogue, la *maïeutique*. Les objets qu'on trouve dans le réel ne sont pas de bons supports pour la pensée, puisqu'ils ne sont que des imitations assez pâles des *Idées* en soi auxquelles l'intellect a accès. Pour pratiquer la philosophie, il est recommandé de se détacher des plaisirs matériels, qui reposent sur des imitations d'*Idées*, et nous détournent de notre but. Il faut également s'abstenir d'écouter les beaux-parleurs, les sophistes, dont le discours enjolive inutilement les idées, qui sont suffisantes exprimées simplement, il ne faut pas non plus compter sur l'art et la



poésie, qui trompent l'esprit. Il faut accéder directement aux Idées par l'exercice du dialogue.

La pensée de la ligne de commande se rapproche donc assez bien de la pensée platonicienne : on y retrouve ce besoin de se passer d'intermédiaire pour accéder à une « véritable » informatique, dépouillée de tout ce qui est inutile (combien de fois l'on a pu entendre des tenants de la CLI traiter les GUI de « gadgets » superflus ou inutiles!).

De plus, la position de Platon face à la poésie est édifiante : il refuse l'allégorie et la métaphore (les principaux ressorts de la poésie) parce qu'il considère qu'elles brouillent l'esprit et l'empêchent d'accéder aux formes intelligibles. En ce qui concerne la ligne de commande, il s'agit également de refuser cela, sur lequel reposent les GUIs : on refuse la métaphore de la page en conservant un texte en blanc sur noir, on refuse les icônes, qui sont toutes également des métaphores d'objets de la vie quotidienne, en leur préférant le texte brut.

La ligne de commande est donc un outil platonicien. Voyons maintenant comment cette pensée s'incarne dans l'outil.

Les applications : des exemples concrets

La philosophie grecque avait pour spécificité de se pratiquer non seulement intellectuellement, mais dans tous les domaines de la vie, jusqu'à former une *esthétique* particulière et cohérente. De la même manière, nous pouvons faire l'hypothèse que tout ce que nous venons d'apprendre sur cet outil s'incarne dans une esthétique bien particulière. Pour illustrer de manière plus complète notre propos, nous allons procéder à l'analyse sémiologique de l'interface de plusieurs programmes, par couple : un sera un programme lancé dans une interface graphique, l'autre sera ou le même programme lancé sans interface, ou son équivalent en ligne de commande.

Commençons par le couple Thunderbird/Mutt, tous les deux des clients mail (*figures 3.1 et 3.2*).

Ce qui saute au yeux de suite, c'est l'inversion des couleurs : Thunderbird est sur fond blanc et Mutt sur fond noir, puisque le premier a intégré la métaphore de la page de texte (du texte noir sur du papier blanc) et l'autre non.



FIGURE 3.1: L'interface de Thunderbird

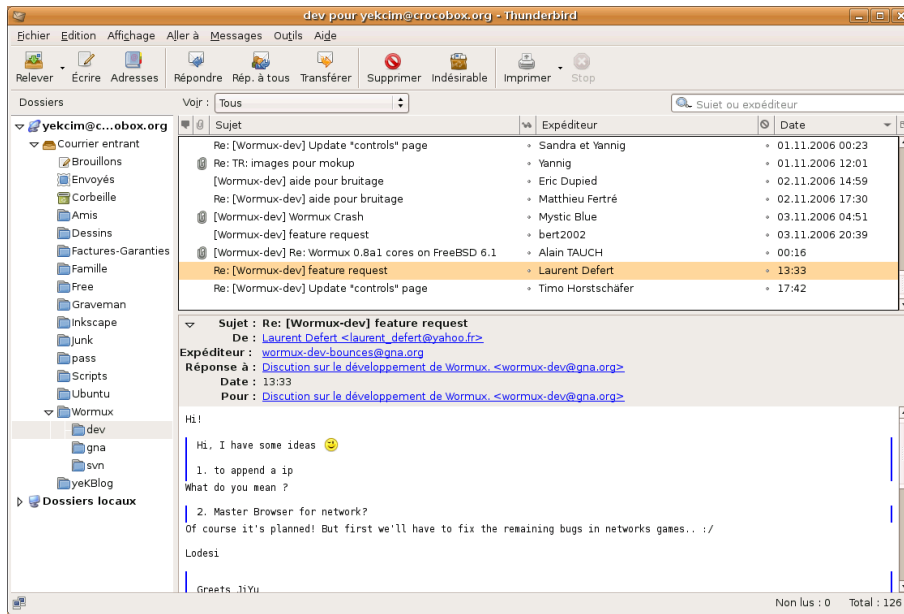


FIGURE 3.2: L'interface de mutt



On se retrouve finalement assez bien entre un programme en graphique et son équivalent en ligne de commande, il sont toujours organisés de la même manière : les mails sont toujours affichés sous forme de liste, quelle que soit l'interface. L'interface graphique permettant l'usage de la souris, tout est fait pour être cliquable, déplaçable avec le pointeur... l'effet de relief signifie à l'utilisateur qu'il peut agir sur l'objet.

Nous remarquons également le minimalisme dont est empreinte l'interface en ligne de commande : alors que l'application graphique propose un espace de prévisualisation du mail, un petit cadre avec la liste des boîtes mail, l'interface de Mutt se



limite à lister les mails entrants. Minimalisme qui fait écho à cette pensée que nous venons d'évoquer, d'une informatique dépouillée, allant le plus précisément et le plus directement au but qu'elle s'est fixée, sans passer par les détours que proposent les interfaces graphiques.

Ensuite, le couple Pidgin/irssi, tous les deux des clients IRC (figures 3.3 et 3.4).

FIGURE 3.3: L'interface de Pidgin, client irc graphique



On remarque que, alors que les interfaces graphiques diffèrent d'une application à l'autre, l'organisation de l'application lancée en ligne de commande est assez similaires : les deux barres (souvent bleues) et haut et en bas, la barre latérale à gauche est également très fréquente. Les applications en ligne de commande se ressemblent esthétiquement beaucoup, ce qui rend difficile pour un observateur externe de savoir ce que la personne est en train de faire, les programmes sont tous faits sur le même modèle.

Ce qui met en lumière une chose, c'est que le *texte* dont dépend l'affichage de toutes les applications en ligne de commande est le même : c'est la configuration du terminal de l'utilisateur. Les applications graphiques dépendent, elles, du gestionnaire de fenêtres. En terme de marge de manoeuvre, soulignons que dans le premier cas



FIGURE 3.4: L'interface de irssi

```

[2:EFnet/#blausoffan (+nps)] Ragnar dockar rattmusféns lillhjárnas fas-plankas á
19 normal]
17:16:56 <firetech> Coper: hade varit trevligt om du kunde ta hand om din
bladserver som står i dsek's serverrum sen geekend HT. :)
17:17:26 <firetech> det är lite trångt på golvet ;)
17:19:02 <Coper> firetech: har jag tänkt också dock blir det ganska jobbigt då
jag inte kommer in där.
17:19:45 <firetech> Coper: bara säg till när du har möjlighet så kan vi nog
fixa det
17:19:58 <Coper> de var inte tanken att den skulle hamna där utan dem missade
nog bara resan till MH. :)
17:20:42 <firetech> Coper: du kommer till DF ikväll?
17:21:05 <firetech> vi skulle kunna fixa det idag isf
17:23:08 <Coper> japp
17:49:35 -!- Jellygoos
[Jellygoose@e4f1e655.31-10-64736c10.cust.bredbandsbolaget.se] has
joined #blausoffan
19:36:47 <@Yarrick> http://www.theregister.co.uk/2009/01/26/dziuba_linux_desktop
/
=====
Day changed to 28 Jan 2009
[ 1:(status) 2:#blausoffan 3:#df.lth.se 4:#rootm 5:#dwww 6:&bitlbee ]
[firetech]
[kontiki] 0*$ irssi [ 0:52]

```

l'utilisateur est tout à fait maître de l'affichage de ses applications, alors que ce n'est pas le cas pour une interface graphique : l'utilisateur *délègue* à son gestionnaire de fenêtres l'affichage plus ou moins joli desdites fenêtres dans lesquelles se trouvent les programmes.

C'est cette proximité d'apparence qui nous conduit à faire l'hypothèse qu'il existe une esthétique propre à la CLI. Ce qui nous confirme dans cette optique est le fait que certains programmes en ligne de commande sont configurés par défaut pour être affichés dans un terminal dont le fond est noir.

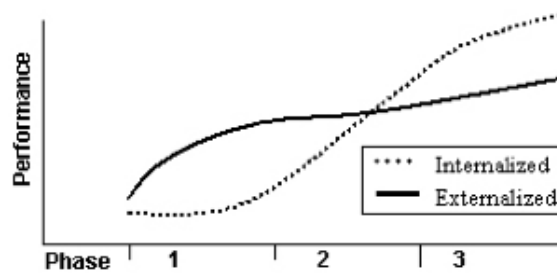
Nous avons bien ici, dans l'image même du texte de ces applications, l'incarnation de la pensée platonicienne que nous avons de développer. En ce sens, on peut bien parler d'une « esthétique de l'efficacité » : dans l'apparence même de ces programmes est inscrite cette pensée. Le choix du vert sur fond noir est éloquent de ce point de vue : il ne s'agit pas d'un choix esthétique au sens premier, mais d'un choix d'économie d'énergie, rien de plus.

Simple jeu d'apparence ou efficacité réelle ?

Mais cette efficacité qu'on s'efforce d'atteindre est-elle aussi difficile à atteindre que les *Idées* de Platon ? Ne sommes-nous pas pris dans une logique de représentation, à arborer tous les signes de la poursuite de l'efficacité, mais sans l'effectuer vraiment ? On peut poser cette question. Nous l'avons déjà un peu évoquée dans la



FIGURE 3.5: Performance comparée des interfaces type GUI (« *Externalized* ») et CLI (« *Internalized* »)



première partie de notre réflexion, lorsque nous essayons de comprendre ce qui faisait que la ligne de commande était vue comme un outil obsolète. Nous avons bien vu que l'obsolescence en informatique n'était pas seulement une question d'interface. Maintenant, nous pouvons retourner la question : est-ce que cette interface est réellement un gage d'efficacité, ou est-ce seulement un signe qui engagerait à croire que c'est plus efficace ainsi ?

La question de l'efficacité est au cœur de nombreux débats concernant le terminal Unix, puisque c'est la qualité qu'on lui revendique le plus. Neal Stephenson en fait l'apologie dans son essai¹⁸. Il en résulte la courbe affichée en *figure 3.7*.

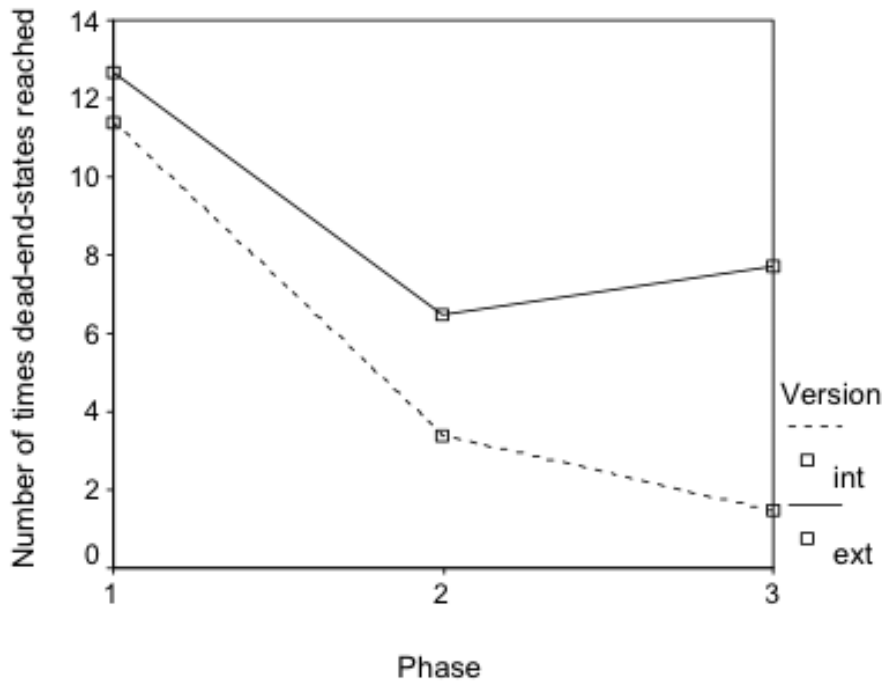
C'est une idée très communément partagée par tous les utilisateurs de systèmes GNU/Linux : les interfaces type GUI seraient tellement centrées sur l'utilisateur, que celui-ci se repose sur elles et réfléchit de moins en moins, alors que les outils du type CLI semblent difficiles à utiliser, mais leur apprentissage payerait si l'on persévère. Les recherches de M. Nimwegen en comportement cognitif par rapport aux interfaces donnent la courbe affichée en *figure 3.7*.

La similitude des deux courbes montre que les études de Nimwegen corroborent donc, étonnamment, la croyance populaire en la meilleure efficacité de la ligne de commande. En tout cas, sur le long terme. En effet, le fait d'« *internaliser* » les informations nécessaires à l'utilisation d'un ordinateur (au lieu de se reposer sur l'interface, c'est à dire « *externaliser* ») stimule plus intelligemment le cerveau. C'est cela qui explique pourquoi l'on devient meilleur au fur et à mesure que l'on utilise la ligne de commande : le cerveau s'entraîne, stocke des informations qu'il n'a plus besoin d'enregistrer par la suite.

18. STEPHENSON, Neal, *In the beginning was the command line*, 1999



FIGURE 3.6: Courbe résultant des études de Nimwegen



Seulement, deux questions restent en suspens : l'interface suffit-elle à être un gage d'efficacité ? N'y a-t-il pas des programmes qui bénéficient d'un jugement favorable alors que cela n'est qu'une question d'apparence ?

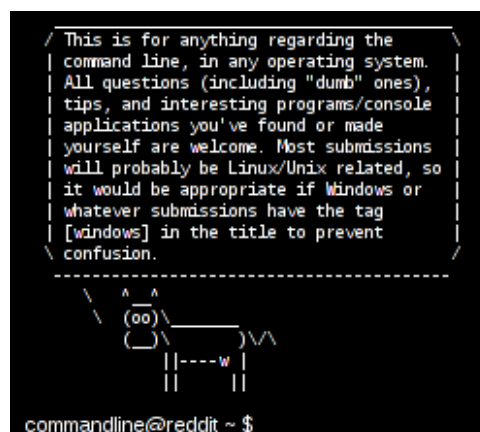
« Il y a des tâches pour lesquelles la ligne de commande est juste efficace. On peut imaginer des interfaces graphiques qui intègrent la ligne de commande. C'est pas forcément lié au terminal pour moi. » remarque une personne interrogée lors d'un entretien. Il souligne que la ligne de commande n'est pas performante pour toutes les tâches que l'on peut remplir sur un ordinateur. Il y a même des tâches qui ne rentrent pas dans son scope : « J'ai eu l'idée de lancer VLC en ASCII pour regarder un match de foot, mais ça n'a pas fonctionné ». De même pour Gimp, l'équivalent de Photoshop, qui nécessite de travailler avec un pointeur. Les inévitables limitations de l'outil ne doivent pas être mises de côté. Nous avons vu que la plupart des utilisateurs de ligne de commande font des compromis pour détourner ces limitations : nous ne pouvons pas rechercher l'efficacité dans tous les domaines avec cet outil.

De plus, à nous focaliser sur une utilisation de la ligne de commande dans une démarche productive, utile, nous oublions en effet que l'efficacité concernant la ligne de commande est toute relative. Il existe un certain nombre de commandes qui ne



FIGURE 3.7: Résultat de la commande *sl*

FIGURE 3.8: La « vache » de Reddit



servent à rien, qui sont juste là pour amuser et faire joli. Par exemple, la commande *sl* qui permet d'afficher... un train en ASCII art dans son terminal (figure 3.7).

De la même manière, le programme *Cowsay* crée des petits dessins en ASCII art, représentant une vache, avec une petite bulle de bande dessinée et un message. Il est possible de le configurer pour avoir un autre animal à la place de la vache, mais la vache est devenue tellement emblématique que sur le forum Reddit dédié à la ligne de commande, elle est affichée à droite, avec un message de bienvenue (figure 3.8)

Si son utilisation est effectivement plus efficace au fur et à mesure que nous en usons, la ligne de commande n'est donc pas en soi un gage d'efficacité. Elle a également des côtés tout à fait contre-productifs. Ce n'est pas un outil si « sérieux » que cela, c'est la manière dont son esthétique est construite qui l'investit de ce sérieux.



Nous savions que la ligne de commande était investie de puissants imaginaires, et qu'il générerait grâce à eux une sociabilité particulière autour de lui. Nous voyons maintenant qu'il est placé au sein d'un réseau d'influences qui contribuent à modifier à la fois notre point de vue sur et notre usage de l'informatique.



CONCLUSION

Répondons donc à la question que nous avons ouverte en introduction : la ligne de commande est une interface particulière en cela que non seulement elle crée une sociabilité autour d'elle, mais elle influence également très profondément la manière de voir et de pratiquer l'informatique.

Nous avons parlé d'*esthétique de l'efficacité* concernant la ligne de commande. C'est en cela que nous pensons que la ligne de commande influence notre attitude face à l'informatique. Si l'on entend cela comme « esthétique de vie », cela nous semble pertinent. Nous pouvons nous rapprocher du travail de Foucault, pour cela : lorsque ce dernier parle de *souci de soi*¹⁹, il évoque la manière dont les Grecs géraient leur existence (à savoir leur intellect, leur vie citoyenne, autant que leur corps et leur sexualité) selon ce qu'on pourrait appeler une esthétique. Il s'agissait de tendre à la vertu autant qu'à un corps sain. Tendre vers le « beau » devait se faire de manière totale. Il y avait chez les anciens une cohérence dans la discipline –le *souci* qu'on s'imposait à soi.

Nous pouvons affirmer que la question de l'utilisation de la ligne de commande est plus une question de choisir un outil qui représente un rapport particulier à l'informatique, une certaine discipline qu'on s'impose face à son outil, que l'on peut très bien rapprocher de l'esthétique de soi grecque. Les Grecs pratiquaient la philosophie autant dans les idées que dans leur manière d'agir en société, de manger et de s'occuper de leur corps. Ici, on a une pensée de l'informatique qui excède la seule question de la performance, et qui vient couvrir des aspects de la vie qui n'ont rien à voir.

« Esthétique » peut également s'entendre au sens premier : c'est pour cela que le point de vue de la présentation de soi nous a semblé particulièrement intéressant,

19. FOUCAULT, Michel, *Histoire de la sexualité 1 : Le souci de soi*, Paris, Gallimard, 1984



surtout avec cet outil qui génère des réactions aussi vives que partagées. L'apparence, l'image et l'image du texte, sont, dans toute interface, déterminants : c'est cela qui donne à l'utilisateur les informations nécessaires à cadrer l'interaction, à savoir quelle marge de manœuvre il possède face à l'interface.

Nous retenons de cette étude que la ligne de commande s'utilise d'une manière bien moins limpide que nous l'aurions imaginé au départ : loin d'une rigidité manichéenne, l'utilisation de la ligne de commande est faite de compromis et d'ajustements divers.

La ligne de commande est donc un outil basé sur et véhiculant une esthétique de l'efficacité par la manière dont il est imprégné d'imaginaires, par la manière dont ces imaginaires sont utilisés et réinvestis par ceux qui utilisent cet outil.

Nous avons essayé de rendre, avec ce travail, un peu plus claire la logique circulaire qui entoure la ligne de commande : on conviendra que l'exposé a pu sembler décousu, mais c'est à dessein : il nous semblait impossible de nous interroger sur les mécanismes par lesquels une interface influe sur un usage particulier de l'informatique sans penser ce phénomène comme étant pris dans une logique de *réseau*. De ce fait, malgré la linéarité de l'exposé imposée par la forme « mémoire », la logique ici est circulaire : pour bien appréhender les interactions entre les différents facteurs, il est nécessaire d'avoir un vue d'ensemble.

Nous conviendrons également que ce mémoire n'explore qu'une petite partie de tout ce qu'il est possible de découvrir à propos de cet outil, notamment en approfondissant les différences qui séparent les principaux systèmes d'exploitation et leur manière de proposer ou non de la ligne de commande à leurs utilisateurs.



ANNEXES

Cette œuvre est mise à disposition sous licence Attribution - Pas d'Utilisation Commerciale - Partage dans les Mêmes Conditions 3.0 France. Pour voir une copie de cette licence, visitez <http://creativecommons.org/licenses/by-nc-sa/3.0/fr/> ou écrivez à Creative Commons, 44 Castro Street, Suite 900, Mountain View, California, 94041, USA.



Glossaire

Interface en ligne et commande et système d'exploitation

Pour commencer, une question doit être posée : où s'arrête l'interface graphique et où commence l'interface en ligne de commande ? De manière générale, on peut définir une interface en ligne de commande comme étant uniquement constituée d'un fond noir et de texte en blanc, où l'utilisateur est invité à écrire des commandes.

Nous avons décidé de limiter notre mémoire à l'analyse de la ligne de commande sous GNU/Linux, pour deux raisons : le système d'exploitation est mieux connu de nous, et de manière générale, même s'il existe de la ligne de commande sous les autres systèmes d'exploitation, c'est sous GNU/Linux qu'elle est le plus utilisée. Comment cela se passe sous GNU/Linux pour qu'un terminal s'affiche à l'écran ?

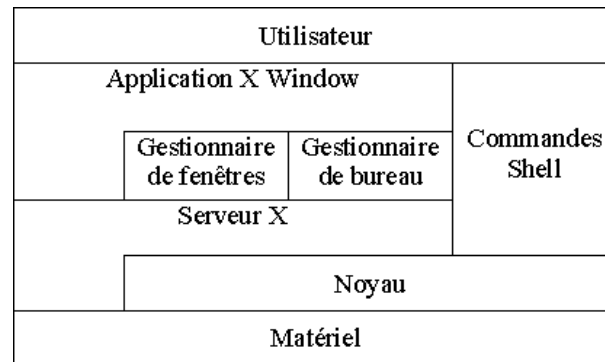
Un ordinateur est conçu de la manière suivante : il y a d'abord les composants électroniques assemblés ensemble, le matériel (la carte mère, le processeur, la carte graphique, la mémoire vive...) qui forment l'architecture technique sur laquelle repose le système. Celui-ci est composé de plusieurs « couches » :

- le noyau Linux, qui gère les interactions avec le matériel ;
- le *shell*, la plus rudimentaire des interfaces utilisateur, *via* laquelle on peut déjà donner des commandes à exécuter à la machine. Le *shell* le plus utilisé sur GNU/Linux est *bash*, mais il en existe d'autres ;
- les programmes en ligne de commande, qui sont utilisables *via* le shell et en augmentent les possibilités ;
- une couche graphique, généralement le serveur X, car peu d'utilisateurs se contentent de la ligne de commande « pure ». Il ne fait qu'afficher ce que les autres programmes lui ordonnent d'afficher ;
- le *gestionnaire de fenêtres*, lui, est chargé de « dessiner » les fenêtres. Eventuellement, un système d'interface graphique plus complet, comme *Gnome* par exemple ;
- les applications X, c'est-à-dire les programmes proprement dits : traitement de texte, lecteur multimédia...

Ce qui peut être résumé par le schéma suivant :



FIGURE 3.9: Architecture des systèmes GNU/Linux



En ce qui concerne les autres systèmes d'exploitation, la place de la CLI dans le système n'est pas tout à fait la même. Si l'architecture système des Mac se rapproche de celle-ci, celle de Windows est tout à fait différente.

Il y a donc deux manières d'utiliser la ligne de commande sous GNU/Linux : ou bien l'on ne fait que lancer un TTY (l'abréviation de « *Text Teletype* »), un *shell* lancé directement sans passer par une interface graphique, ou bien l'on lance ce qu'on appelle une console ou un terminal, depuis une interface graphique en général un gestionnaire de fenêtres (*window manager* dans la plupart des distributions GNU/Linux).

Maintenant que ces principes sont posés, il nous faut faire une distinction entre ce qu'on appelle en informatique le *shell* et la ligne de commande. Le *shell*, à proprement parler, est, comme son nom l'indique, une « enveloppe » : c'est un programme qui permet à un utilisateur d'interagir avec le *noyau*, mais il n'est pas forcément en ligne de commande. Il existe des *shells* ayant une interface graphique.

Ce mémoire est donc focalisé sur le *shell* en ligne de commande, et ce que l'on peut lancer en ligne de commande, donc, à l'intérieur de ce dernier.

Interface en ligne de commande et interface graphique

Les interfaces graphiques sont apparues de manière concomitante avec la souris, et les premiers systèmes d'exploitation grand public. Elles correspondaient à un besoin, qui était de commercialiser des ordinateurs à grande échelle, à un public qui n'y était pas forcément familier. Il fallait lui simplifier au maximum la tâche, pour qu'il n'ait pas d'apprentissage supplémentaire à faire. Les GUIs, de ce fait, se basent sur des métaphores connues, telle que celle du « bureau » des « dossiers » ou de la « corbeille ».



Aujourd'hui, la plupart des systèmes d'exploitation comportent un GUI, même les distributions GNU/Linux comme Debian ou ArchLinux. Le GUI sous GNU/Linux est géré par deux applications, qui sont, d'une part, X.org, qui apporte les éléments d'interface graphique, et le *window manager*, dont le travail est de « dessiner » les fenêtres à afficher selon les besoins de l'utilisateur.

Interface en ligne de commande et programmes en ligne de commande

Ce que nous appelons ici « programme en ligne de commande » est un programme que l'on peut lancer dans un terminal ou un TTY, et qui obéit à une certaine esthétique propre à la ligne de commande. De nombreux programmes peuvent se lancer en invoquant simplement leur nom dans un terminal, si bien que l'on peut utiliser son ordinateur pour des tâches quotidiennes comme écouter de la musique, relever ses mails, écrire du texte... tout en ligne de commande.

Nous séparons ici interface et programmes car les deux ne sont pas conçus dans la même optique. En effet, le fait qu'il existe une interface en ligne de commande, ou à minima un émulateur de terminal ne veut pas forcément dire qu'il y aura des applications en CLI. En revanche, un programme développé pour la ligne de commande a été pensé uniquement pour cela. Il y a un choix à la base de la conception de l'application. Elles sont pensées pour une certaine esthétique et un certain univers, bien différent de celui en GUI.

Ce qu'on peut remarquer d'emblée, c'est qu'on parle d'interface en ligne de commande, mais en réalité, il existe très peu d'interfaces complètement en ligne de commande, à part le TTY, qui est très peu utilisé, au final. En effet, la plupart des ordinateurs qu'on désigne abusivement comme étant configurés avec une interface en ligne de commande sont en fait configurés avec un GUI fait pour un utilisateur qui ne lance que des commandes. Peut-on, à ce moment, opposer aussi nettement CLI et GUI? Nous distingueront donc plutôt une utilisation de l'informatique orientée vers les GUI d'une autre, orientée vers la CLI.

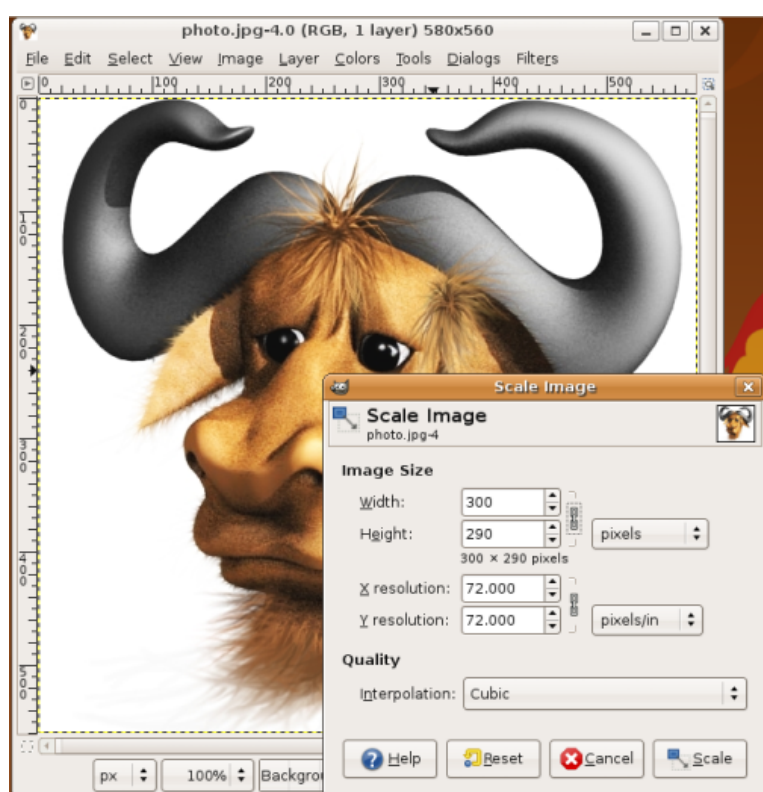


Put Yourself In Command

Commands enable you to do all sorts of powerful things on your computer. We'll demonstrate this by looking at an everyday task that might be familiar to you. If you use a digital camera, you probably have a folder full of images on your computer. Imagine you wish to resize the image `profile.jpg` so it is 300 pixels wide and save it as a new image called `profile_small.jpg`.

Using an image editing software, the steps you need to go through might look as follows:

1. Open an image editor from the Applications menu.
2. Click the *File>Open* menu item.
3. Browse to the folder with your images.
4. Click the image file `profile.jpg` and then *Open*.
5. Click the *Image>Scale Image* menu item, for resizing images.
6. Change the image Width to 300 pixels and click *Scale*.
7. Click the *File>Save As* menu item to save the file.
8. Enter `profile_small.jpg` as the new file Name and click *Save*.



Using the command line, you can achieve the same operation by typing:

```
convert -resize 300 profile.jpg profile_small.jpg
```

That's 1 step in the command line instead of 8 steps using the graphical interface. Perhaps you think that the time you spend reading this book and studying the commands isn't worth saving 7 steps. But what if there were 30 images to be resized. Do you still want to open each image individually and repeat the process 30 times using the image editor? That would total 240 steps that you have to go through. Wouldn't you rather just type one command instead and get the job done?

A single command could do the same thing whether you had 30 images, 300 or 3000. This is one of the most compelling reasons to start using the command line. You might start with a slow learning curve, but in the long run, it will save a lot of time. Even more important, learning the command line will open up interesting



possibilities and fun ways of working. Lets look at some more reasons why learning the command line is a good idea.

Summarizing the advantages of using commands

Many people who give the command line a try are so amazed by its possibilities that they do not even want to go back to a graphical user interface! Why? Well, in brief, the command line offers the following main advantages over common graphical software:

- **Flexibility.** With graphical programs you sometimes hit a limit, finding that you just can't do what you want or need to find cumbersome work-arounds to program limits. Working on the command line, though, you can combine commands to yield a virtually infinite range of new and interesting functions. By combining commands creatively, you can make the command line do exactly what you want -- it puts you in control of your computer.
- **Reliability.** Graphical programs are often immature or even unstable. In contrast, most of the tools that the command line offers are highly reliable. One of the reasons for this is their maturity: the oldest command line programs have been around since the late 1970s, which means that they have been tested for three decades. They also tend to work the same way across different operating systems, unlike most graphical tools. If you want a Swiss Army knife you can rely on, the command line is for you.
- **Speed.** Fancy graphics eat up a lot of your hardware's resources, often resulting in slowness or instability. The command line, on the other hand, uses the computer's resources much more sparingly leaving memory and processing power for the tasks that you actually want to accomplish. The command line is also intrinsically faster: instead of clicking through long chains of graphical menus, you can type commands in a dozen or so keystrokes, and often apply them to multiple files or other objects. If you are a fast typist, this will enable you to drastically increase your productivity.
- **Experience.** Using the command line is a great learning experience. When you use the command line you communicate with your computer more directly than with graphical programs, thus learning a lot about its inner workings: using the command line on a regular basis is *the* way to becoming a GNU/Linux guru.
- **Fun.** Have you ever wanted to be like those cool computer hackers who can make a GNU/Linux machine do things that you haven't even dreamed of? Almost all of them do that using the command line. Once you have learned how to use this powerful tool, you will find yourself doing fun and interesting stuff that you hadn't even thought to be possible.

The Value of Scripting

But wait, there's more! You can also store commands in text files. These text files are called *scripts* and can be used instead of typing out a long series of commands each time. For example, if you store commands in a file called `mycommand.sh`, you don't have to type out the commands again, but simply type :

```
mycommand.sh
```

Additionally you can combine commands together in simple or sophisticated ways. Further, you can schedule scripts to occur at a specific time or date or at the occurrence of a specific event on your computer.

You can also write scripts so they accept additional information from you. For example, an image resizing script might ask you what size the images should be resized to before it starts.

Ever tried to do anything remotely like that by using a Graphical User Interface? Perhaps now you can see how working with the command line interface starts to open a whole new world to using your computer.

Cette œuvre est mise à disposition sous licence Attribution - Pas d'Utilisation Commerciale - Partage dans les Mêmes Conditions 3.0 France. Pour voir une copie de cette licence, visitez <http://creativecommons.org/licenses/by-nc-sa/3.0/fr/> ou écrivez à Creative Commons, 444 Castro Street, Suite 900, Mountain View, California, 94041, USA.



Is my computer sick?

Another use for the command line is to check the well-being of your computer. There are many commands you can use to check every facet of your computer's health, from the amount of space left on the hard drive to the temperature of the CPU. If your computer is acting poorly and you don't know what the matter is, a few commands will help you quickly determine whether it's a hardware or software issue, and help you make a quick recovery.

Spanning the network

There is one other interesting feature of command line interfaces that GUIs can't match: interaction over a network. Imagine you have a computer in another room and you wish to turn it off. How do you do that? Easy, right? Get up, walk to the computer and click on the 'shutdown' button.

Well, those that know how can connect to the computer in the next room via the command line and type *halt*.

That might seem trivial. Perhaps it's actually better for you to get up off that comfy chair and expend 5 calories walking to the next room. However, what if the computer you wanted to shut down was in another suburb? In another city? Another country? Remote control of that computer might then be very useful.

Shutting down a remote computer is just a start. Everything you can do on the command line you can do on the remote computer. That means you can run scripts, execute commands, edit text files, check the diagnostics, etc. The world of the command line just got a whole lot bigger.

Even graphical programs are commands

When you click on an icon or menu item to start a program, you're running a command. You may find times when it's useful to understand what command you're running. For instance, if you suspect a program is running invisibly in the background and slowing your computer, you can find its command and terminate the program. GUI programs often send more error messages to the CLI, than they show in dialog boxes, this is useful to diagnose problems.



Découverte de la ligne de commande

Vincent-Xavier Jumel,
Aurélien Couderc

Présentation

Les shells textuels

Introduction au shell textuel

L'invite de commande

Lancement d'un programme

Exemple

Complétion des commandes

Formation d'une commande

Navigation dans le système de fichiers

Le système de fichiers

Navigation

Où suis-je ?

Chemins particuliers

Complétion des chemins

Les permissions sur les fichiers

Lecture des fichiers

Quelques programmes de

Trouver de l'aide

man

- 1 Commandes utilisateur
- 2 Appels système
- 3 Fonctions de bibliothèque
- 4 Fichiers spéciaux
- 5 Formats de fichier
- 6 Jeux
- 7 Divers
- 8 Administration système
- 9 Interface du noyau Linux



- Découverte de la ligne de commande
 - Vincent-Xavier Jumel, Aurélien Couderc
- Présentation
 - Les shells textuels
 - Introduction au shell textuel
 - L'invite de commande
- Lancement d'un programme
 - Exemple
 - Complétion des commandes
 - Formation d'une commande
- Navigation dans le système de fichiers
 - Le système de fichiers
 - Navigation
 - Où suis-je ?
 - Chemins particuliers
 - Complétion des chemins
 - Les permissions sur les fichiers
- Lecture des fichiers
 - Quelques programmes de

L'invocation normale de la commande `man` prend comme arguments le numéro de section et le nom de la commande. Si le numéro de section est omis, la première section comportant une page de manuel sur la commande sera utilisée.

Les pages de manuel s'affichent par défaut dans un logiciel de pagination, comme `more` ou `less`

```
user@hôte:~$ man ls
```

 affiche l'aide de la fonction `ls`.



Un aide mémoire

De nombreux programmes supportent l'option spéciale `--help`, `-h` ou `-help` qui fournit un aide-mémoire sur le terminal.

```
user@hôte:~$ ls --help
```

Découverte de la ligne de commande

Vincent-Xavier Jumel,
Aurélien Couderc

Présentation

Les shells textuels

Introduction au shell textuel

L'invite de commande

Lancement d'un programme

Exemple

Complétion des commandes

Formation d'une commande

Navigation dans le système de fichiers

Le système de fichiers

Navigation

Où suis-je ?

Chemins particuliers

Complétion des chemins

Les permissions sur les fichiers

Lecture des fichiers

Quelques programmes de

NAME

man – an interface to the on-line reference manuals

SYNOPSIS

```
man [-C file] [-d] [-D] [--warnings[=warnings]] [-R encoding] [-L locale] [-m system [, ...]]
[-M path] [-S list] [-e extension] [-i|-I] [--regex|--wildcard] [--names-only] [-a] [-u]
[--no-subpages] [-P pager] [-r prompt] [-7] [-E encoding] [--no-hyphenation] [--no-justification]
[-p string] [-t] [-T [device]] [-H [browser]] [-X [dpi]] [-Z] [[section] page ...] ...
man -k [apropos options] regex ...
man -K [-w|-W] [-S list] [-i|-I] [--regex] [section] term ...
man -f [whatis options] page ...
man -l [-C file] [-d] [-D] [--warnings[=warnings]] [-R encoding] [-L locale] [-P pager] [-r
prompt] [-7] [-E encoding] [-p string] [-t] [-T [device]] [-H [browser]] [-X [dpi]] [-Z] file ...
man -w|-W [-C file] [-d] [-D] page ...
man -c [-C file] [-d] [-D] page ...
man [-hV]
```

DESCRIPTION

man is the system's manual pager. Each *page* argument given to **man** is normally the name of a program, utility or function. The *manual page* associated with each of these arguments is then found and displayed. A *section*, if provided, will direct **man** to look only in that *section* of the manual. The default action is to search in all of the available *sections*, following a pre-defined order and to show only the first *page* found, even if *page* exists in several *sections*.

The table below shows the *section* numbers of the manual followed by the types of pages they contain.

- 1 Executable programs or shell commands
- 2 System calls (functions provided by the kernel)
- 3 Library calls (functions within program libraries)
- 4 Special files (usually found in */dev*)
- 5 File formats and conventions eg */etc/passwd*
- 6 Games
- 7 Miscellaneous (including macro packages and conventions), e.g. **man(7)**, **groff(7)**
- 8 System administration commands (usually only for root)
- 9 Kernel routines [Non standard]

A manual *page* consists of several sections.

Conventional section names include **NAME**, **SYNOPSIS**, **CONFIGURATION**, **DESCRIPTION**, **OPTIONS**, **EXIT STATUS**, **RETURN VALUE**, **ERRORS**, **ENVIRONMENT**, **FILES**, **VERSIONS**, **CONFORMING TO**, **NOTES**, **BUGS**, **EXAMPLE**, **AUTHORS**, and **SEE ALSO**.

The following conventions apply to the **SYNOPSIS** section and can be used as a guide in other sections.

| | |
|---------------------------|--|
| bold text | type exactly as shown. |
| <i>italic text</i> | replace with appropriate argument. |
| [-abc] | any or all arguments within [] are optional. |
| -a -b | options delimited by cannot be used together. |
| <i>argument</i> ... | <i>argument</i> is repeatable. |
| [<i>expression</i>] ... | entire <i>expression</i> within [] is repeatable. |

Exact rendering may vary depending on the output device. For instance, **man** will usually not be able to render italics when running in a terminal, and will typically use underlined or coloured text instead.

The command or function illustration is a pattern that should match all possible invocations. In some cases





Note: The author has reserved the right to be wrong! :)

10 Reasons Why the Command Line is More User-Friendly than the Desktop

Date/Time Permalink: [11/24/07 08:00:38 am](#)

Category: [LINKS and Lists](#)

I aim all of these not just at Linux, BSD, and Unix-alikes, but at every computer system ever. Nor do I say all of this as a power-using geek, but to apply it to every computer user everywhere.

1. Keying is faster than mousing.

You might argue this point, and if you do, I'd just point out "Why are there keyboard shortcuts on most menu entries in a GUI?" And why are they called "shortcuts"? Because using a mouse to do something is the long way to do it, each and every time.

2. It's easier to both give and get help.

Nearly every Debian system has synaptic, and I'll bet most people use it. Yet even amongst synaptic users, they'll still answer questions with an "apt-get" statement. It doesn't matter if you use apt-get or synaptic; it's still more efficient to use the command expression behind the GUI action to answer questions online. You can interpret the answer into the GUI tool of your choice. Ever seen somebody answer a question that has no command-line answer? "Go here, double-click that, click this pull-down, click the arrow, click 'yes', click 'edit', double-click this, that, highlight this..." And this applies on Windows as well; even there it's easier to define an action in DOS commands than it is to describe mouse actions.

This is why our mail is addressed to our house, instead of having instructions to the postman: "Go down main street, turn left, go to spruce street, turn right, go to the eighth building on the right, go up the stairs..."

3. Repetitive stress injury comes from the mouse, not the keyboard.

Do something that uses the keyboard for awhile, then do something that uses the mouse for awhile. How do your hands feel? I especially notice that when I play a game like Diablo or Starcraft, my right hand is cramping terribly after a few hours. But I can play Nethack all day and not feel any pain at all. It stands to reason that if you filter the work of both hands and ten fingers through one wrist and three fingers, you'll have more repetitive motion stress.

4. Commands are standard where GUIs are not.

How do I find a word in a document using a GUI? Do I click "edit - find and replace"? Do I click "view - find"? Do I click "find - search page"? Does the dialog pop up, or does it pop in from the side, or does it show up in a bar at the bottom? Do I have to click close afterwards, or will it close itself? Is the 'find' dialog separate from the 'replace' dialog, or are they combined? Does the dialog on the program understand a regular expression? It all depends on which program you're using - and often which version. And - again, no matter what platform - there's a dozen possible GUI tools to do the same thing with complete feature overlap. Which one do you have available?

How do I find a word in a document using a command line? Grep! I learned grep once in 1998 and haven't had to open the man

Cette œuvre est mise à disposition sous licence Attribution - Pas d'Utilisation Commerciale - Partage dans les Mêmes Conditions 3.0 France. Pour voir une copie de cette licence, visitez <http://creativecommons.org/licenses/by-nc-sa/3.0/fr/> ou écrivez à Creative Commons, 444 Castro Street, Suite 900, Mountain View, California, 94041, USA.



page since. I'll still be able to use grep 20 years from now. Every Unix-based system has it, even Macs and Windows has it available, and it's always the same from platform to platform.

5. **Commands are more powerful.**

I don't care how well-designed the GUI is, there's always something you need to do that they forgot to put in a menu item for. But commands can be adapted to many purposes. Especially when you have to do the same action many times. Go ahead, resize a bunch of images to thumbnails in your GUI tool; you'll be working all day while I can type out an Image Magick incantation in one minute and go take a nap while it does it all automatically.

How about editing a document where you need to change all instances of something, *except* where a special case is true? Unzip a stack of downloaded files, rename them all to a specific scheme, and move them all to different directories? The GUI is good only for limited-use cases and specific actions, but it only takes you that far.

6. **You can automate commands.**

Really, you only have to learn each command once. Then, no matter how complex, you can save it in a shell script, run it from a menu item, execute it from one key-press, or even give it to cron to perform automatically. There is no computing job on Heaven or Earth that cannot be reduced to a single motion, given a proper scripting environment.

7. **Commands are easier to remember than clicks.**

Face it, our brains handle words better than they handle spacial sequences of motions. How many times have you tried to show somebody else how to perform a mouse task, only to realize that without sitting in front of the machine doing it yourself, you can't remember how it goes? Learn a command and it's yours for life.

8. **GUIs are always changing.**

Everybody wants to re-invent the widget set, patent the scroll wheel, re-think the dialog, put their own stamp on the GUI environment. Bring up GUI design, and there's a flame war all over the place about simple things like where a menu should go and whether to use a spin-box or a combo-box. Commands have been performed the same way since the keyboard was invented, end of story.

9. **Commands are more error-safe.**

All I ever hear is how you can accidentally type "rm -rf /*" or "del *.*" in the wrong place and wipe out your file system, but look at practical experience. How many times have any of us actually done that? Now, how many times have you clicked 'yes' or 'allow' to the wrong dialog, clicked the wrong button and closed a window when you meant to minimize it, opened the wrong file in a dialog, selected the wrong action in a menu, had to make three tries to drag a selection box around a group of files, dragged and dropped a folder to the wrong place... the list goes on and on. Mousing takes intense co-ordination and concentration to hit a target a few pixels wide, and in the event of a mistake the wrong action is always performed instantly.

A command line does not fire off until you hit enter. So you can read it after you've typed it to be sure it's what you want to do. You can edit it before you execute it. And your brain is actually engaged in the task, instead of focusing on performing millimeter-precise actions with a few muscles in one hand.

10. **You have to type to use a computer anyway.**

You can get quite far on a computer without a mouse, as long as the programs are set up to accept keystroke commands. But try this: unplug the keyboard and hide it. Now, how far did you get using the computer? How can you IM, email, edit, comment, name files? No matter how much you love your GUI interface, you still spend the bulk of your time using a keyboard.



UPDATE: Normally, I don't comment on who links to what. But seeing as how this post has gotten a lot of attention, I figured I'd mention something. First, while I'm here, as always, thank you all for your interesting feedback - even if you disagree with me!

Cette œuvre est mise à disposition sous licence Attribution - Pas d'Utilisation Commerciale - Partage dans les Mêmes Conditions 3.0 France. Pour voir une copie de cette licence, visitez <http://creativecommons.org/licenses/by-nc-sa/3.0/fr/> ou écrivez à Creative Commons, 444 Castro Street, Suite 900, Mountain View, California, 94041, USA.



Questionnaire établi pour la première salve d'entretiens

Ce questionnaire était plus un guide d'entretien qualitatif qu'un véritable questionnaire, c'est pourquoi les questions n'ont pas été reportées dans les notes

Infos générales

1. Genre
2. Âge
3. Études

L'interrogé et son rapport à l'informatique

1. Combien de temps passes-tu en moyenne sur ton ordinateur par jour ?
2. Sous quel système d'exploitation es-tu ?
3. Si tu es sous Linux, sous quelle distribution ?
4. Qu'est-ce qu'un logiciel libre ?
5. En utilises-tu ?
6. Que fais-tu lorsque tu es sur ton ordinateur ? Quels programmes lances-tu ?

L'interrogé, l'informatique et les gens

1. As-tu des geeks dans ton entourage ?
2. Te considères tu comme geek ?
3. Geek/nerd ?
4. Tes proches comprennent-ils ce que tu fais quand tu travailles sur ton ordinateur ?
5. Parles-tu d'informatique avec tes proches ?
6. Troll ?

L'interrogé et la ligne de commande

1. Que penses-tu de l'interface en ligne de commande ?
2. Que penses-tu des interfaces graphiques ?
3. Lequel des deux utilises-tu ? Pourquoi ?
4. L'avantage de LaTeX par rapport à un éditeur de texte graphique ?
5. Term nazi / kikoo ?



Notes du premier entretien

Garçon 21 paris

Étudiant à Sciences Po (master droit économique, stage césure)

Laptop 8h par jour au moins pour le boulot, puis chez moi.

10h de sommeil, je ne sais pas...15h par jour.

Si je ne sors pas je suis toujours sur mon ordinateur

Pour faire quoi ? Tout.

Écrire, lire des mails, après, des articles à lire, un peu de blog, puis des trucs pour le boulot. Écrire, quoi.

J'utilise la ligne de commande parce que j'utilise pas flash.

Je l'utilise pour tout ce qui est apt-get, un tas de trucs, des petits programmes que je préfère en ligne de commande.

Subversion, Nano....

Je pourrais le faire en graphique, mais j'sais pas, Nano, je l'utilise juste quand j'ai besoin de noter un morceau de texte, je vois pas pourquoi, j'aurai besoin d'option en plus.

En réalité, je lance la ligne de commande à Sciences Po quand il y a des gens derrière moi pour leur faire peur.

Une fois j'étais en amphi, le cours était pas intéressant. Je commence à lire un truc en arabe. Pour lire en arabe je regarde tous les liens :D

Je lis ça, et après j'ai ouvert mon terminal, et les gens derrière moi ont fait « qu'est-ce que tu fais ? ». C'est marrant, je ne vois pas le rapport.

Je faisais un truc un peu bizarre pour eux. On leur a jamais montré ...Quand j'ai commencé Ubuntu, je faisais pas du tout en ligne de commande et c'est arrivé un peu par accident, et j'ai fini par me dire, ah c'est mieux.



C'est propre au logiciel libre à mon avis. C'est les seuls systèmes qui vont te permettre de passer à la ligne de commande assez vite. Regarde Mac, c'est un Unix, ils peuvent passer beaucoup de temps sans utiliser un terminal, ils ne connaissent pas leur mot de passe root.

Le terminal est là, accessible, et y'a beaucoup de gens qui sont prêts à te montrer comment ça marche..

A partir du moment où ta tâche principale c'est travailler du texte, y'a pas mieux que le texte.

Je pense à mutt, par exemple.

C'est un bon exemple : parce que pour te lancer, t'es un peu lâché dans la nature.

L'aspect look, j'essaye de me rapprocher des interfaces de Tron Legacy. Je suis allé jusqu'à éditer le gnomeshell.css.

J'ai pris les screenshots du site, et j'ai recopié.

Mon terminal c'est Quakee, c'est un drop down, qui se superpose à ton bureau, c'est classe, en plus c'est transparent, ça fait un peu hacker.

Dans le 2, dans l'article, le mec a décidé de mettre de vraies commandes. Ils utilisent un vrai emacs avec des vraies commandes. C'est juste que les développeurs de Gnome ont des goûts de chiotte :D

Ce qui est rigolo, c'est qu'on a des restrictions qui n'ont plus lieu d'être dans le terminal. On continue à envoyer des mails cassés à 72 caractères alors que les écrans ont grandi.

Histoire sur Reddit : le mec dans un café, il ouvre un terminal pour faire quelque chose de tout simple, et il y avait un policier derrière lui. Il le menotte et tout. Il était gêné quand il a compris qu'il n'y avait rien.

Dans l'imaginaire populaire, on associe la ligne de commande à faire des trucs dangereux. Au final on a peur des gens qui savent utiliser la technologie. En tout cas ça impressionne pas les filles...

Y'a des tâches pour lesquelles la ligne de commande est juste efficace. On peut imaginer des interfaces graphiques qui intègrent la ligne de commande.

C'est pas forcément lié au terminal pour moi.

La ligne de commande c'est juste un type d'interface.



J'ai eu l'idée de lancer VLC en ASCII pour regarder un match de foot mais ça n'a pas fonctionné.

Vu qu'Apple est un peu présenté comme l'interface parfaite, ce serait intéressant de voir les clash entre Raskin et Jobs à l'époque.

Regarde, au Fosdem, il y a de plus en plus de mac.

Notes du second entretien

Garçon / 30 ans / administrateur système

Entre 8h minimum à 16h d'ordi par jour. Pas vraiment de règles, moyennes.

Téléphone : je ne téléphone pratiquement jamais, donc c'est comme un ordinateur.

Si tu comptes le téléphone, je suis tout le temps devant un écran sauf quand je dors.

Archlinux i680

Assez peu de jeu, de multimédia, globalement je bricole : écrire des scripts, du logiciel, s'informer beaucoup. Les quelques réseaux sociaux que je fréquente...

Pour le jeu, j'ai une console de jeu pour ça.

Beaucoup de discussion avec des gens divers et variés. Je ne suis pas mono-média.

Je travaille avec des développeurs qui écrivent du code toute la journée. Ils vont pas essayer de comprendre comment ça marche ou le customiser. Ils l'utilisent comme un outil.

Un ordinateur c'est quelque chose qu'est fait pour traiter de la donnée. C'est une interface homme / données.

Même en travaillant avec des développeurs, quand on voit mon ordinateur, ils ont peur. Ils prennent un ordinateur tel qu'il est, ils ne se sont jamais posé la question d'adapter l'outil à l'usage.

Quand t'es sous windows ou sous mac, rapidement tu passes à autre chose. C'est comme si on te donne un marteau on te demande d'enfoncer une vis, quelqu'un va te dire qu'il veut pas le faire, moi je vais essayer de le faire de toute façon. C'est une question d'adapter son outil à soi. C'est plutôt les hackers qui font ça, ouais.

Après ça dépend, mais t'as des entreprises qui te montent des outils de la mort et t'es obligé d'installer Outlook.



Je ne pense pas qu'il existe des ordinateurs utilisés de la même façon dans le monde. C'est un peu un reflet de ce qu'on est.

Dès qu'on arrive à des outils modulaires...les gens ont des idées précises de ce qu'ils veulent obtenir. Un bon outil doit faire ce que je veux qu'il fasse, sans me poser trop de questions.

Il y a beaucoup d'interfaces mal conçues. Prenons Unity. Ils veulent faire une interface qui ne viennent là que quand tu as besoin, c'est leur but.

J'utilise la ligne de commande et des windows managers extrêmement simples, c'est que j'ai juste 15px d'info en haut et en bas.

Je n'utilise pas un ordinateur parce qu'il est joli, sinon je serai sous mac.

Il se penche sur l'ordinateur de sa compagne, sous Ubuntu, avec la dernière interface graphique : Là pour chercher une application, je galère.

L'interface ne se met pas en milieu de ton chemin.

Avec une ligne de commande, tu as une courbe d'apprentissage plus longue.

Avec une interface graphique, tu apprends en te perdant : et tu vas devoir refaire la même chose.

Une interface, ça répond à « comment je fais pour faire ça ».

Le fond d'écran, 90% du temps, ça ne me sert à rien. Même Firefox, t'as plein de place perdue, la barre en haut ne sert à rien. Là en fill screen c'est intéressant.

L'interface en ligne de commande, une fois que tu sais, c'est efficace très rapidement. Tu décris ce que tu veux, par exemple pour convertir un fichier, ça me prend 5 secondes.

Une interface graphique qui est bien foutue s'adapte à son utilisateur (ce qui est le cas en ligne de commande parce que tu finis par faire des scripts et des alias).

Le problème des GUI, c'est qu'ils me forcent à me poser une question : comment est ce que je peux faire ce que je veux ? En ligne de commande, tu lis le man, une fois que tu sais tu le fais. Tu te poses moins cette question.

Les GUI le problème, c'est qu'on a voulu faire des interfaces pour tout le monde, du coup on en fait pour personne. Il faut faire des interfaces qui apprennent de leurs utilisations.



Le but d'une interface c'est de présenter les données dont j'ai besoin quand j'ai besoin, comme j'ai besoin, et non pas d'apprendre comment une interface va me présenter des données.

Je ne dis pas qu'une interface ne nécessite pas d'apprentissage. En ligne de commande tu as 36 manières de faire la même chose tu finis par faire comme tu veux.

Il y a des interfaces tactiles, on peut toucher. Mais on utilise dix pourcent de ce qu'on fait avec nos mains. L'interface de Minority Report ne gêne pas pour faire ce qu'il a à faire, c'est pas du pur tactile, il fait des mouvements.

Je manipule essentiellement du texte. Ça se parse facilement. Faire un ctrl + F sur une image, je ne sais le faire. On peut le faire, mais si on a produit une image.

C'est vraiment du texte : c'est historiquement le plus rapide, le plus facile : on a affiché du texte avant d'afficher des images. Dans une civilisation du pictogramme comme en Chine, je ne suis pas sûr qu'ils aient fait les ordinateurs pareil.

Les ordinateurs, c'est des machines de Turing. L'ordinateur n'a pas évolué depuis 1940. On a juste mis plus de puissance.

Quand tu regardes, les interfaces n'ont pas autant évolué que ça. On a juste fait des icônes plus jolies.

Apple, ça me gave. Je ne peux pas éditer mon fichier de conf à la main.

Le noir c'est de la lumière qui n'émet pas, ça n'abîme pas les yeux. Il faut que ce soit confortable pour le temps que j'y passe dessus. C'est comme les jeux vidéos. Je peux passer à jouer à Dungeon Crawl pendant des heures.

Peu importe la forme si le fond est intéressant. C'est comme un film en 3D et le scénario. Sur les interfaces, on vend du bling-bling.

Mais on ne contrôle pas la machine. « if you can't open it, you don't own it ».

Ubuntu et Mac, tu as une application, il faut la chercher. Le problème de Unity, c'est que la documentation...

je perds du temps dans une interface graphique. Parce que ce n'est pas moi qui l'ai conçue.

J'ai besoin de deux mains sur un clavier, c'est une perte de temps d'attraper une souris.



Pourquoi quelqu'un se fait chier à faire sa vidange soit-même ? Parce qu'il veut que ça doit être bien fait.

IL y a des gens pour qui l'informatique est une contrainte. Il vont penser que ça ne marche pas et seront confortés dans cette idée. C'est une prophétie autoréalisatrice.

La ligne de commande a un côté universel : partout, ls va fonctionner. Ceci dit, j'utilise des lignes de commandes bizarres pour faire peur aux gens. Le truc, c'est que est ce que je peux faire ça en ligne de commande, en fait, oui, donc tu le fais. Le term Nazi : optique trollesque.

Dans un serveur, il n'y a pas d'interface graphique, je suis obligé de le faire. Une interface graphique sur un serveur, c'est beaucoup pour pas grand chose. Après c'est bien de voir jusqu'où je peux aller.

Le Kikoo, c'est Jean-Kévin. Ça vient des mecs qui ponctuent leur phrases par LOL, font kikoo lol asv. Ils génèrent plus de bruit que d'info dans leur signal. Il frime parce qu'il sait utiliser trois commandes mais ne sait pas vraiment.

On est dans un monde de signal. Dans la Théorie de l'information : signal = bruit / information. Une information va modifier mon état. Les interfaces te permettent de traiter la différence entre bruit et information. On doit réduire le plus en plus le signal. Soit tu écrèmes du bruit et tu gardes l'info soit tu fais l'inverse On veut faire une interface jolie avant d'être efficace, donc on privilégie le bruit par rapport à l'information. Wireshark, c'est une bonne interface : tu te poses pas la question trois mois.

Il y a des utilitaires en ligne de commande qui sont inutilisables. Gimp, dans un terminal, je ne vois pas l'intérêt, tu travailles avec un pointeur.

Notes du troisième entretien

M / prof de maths / 28

Beaucoup trop. Un minimum de trois heures par jour. C'est les jours normaux où je ne me dis pas aujourd'hui, je ne bosse pas sur mon ordi.» là ça explose...

La spécificité, c'est que je ne passe pas de temps devant mon ordi professionnellement, je ne sors que mon ordi à la fin des cours pour noter mais c'est tout.



Je suis sous GNU/Linux, bien sur. Oui, j'utilise la ligne de commande. Je suis venu à l'utiliser à travers ma formation à l'école d'ingé. On nous a mis sur des terminaux. On a eu un TP « utiliser la ligne de commande sous Linux ». On passait notre temps sur des terminaux Sun Solaris tous pourraves. Pour avoir la même chose partout, y'avait que la ligne de commande pour arriver à ça. Et puis je me suis aperçu que je pouvais utiliser la ligne de commande pour me connecter en ssh à des machines plus puissantes. C'était minimaliste, on appris ls, pwd, des commandes simples...et ça s'est fini avec deux commandes géniales, qui s'appellent man et info.

J'avais récupéré un vieux pc chez mes parents, un vieux bazar immonde. C'est le moment où j'ai découvert le logiciel libre, Mandriva, Debian, etc. Et je voulais expérimenter. C'était une brouette.

Les installations plantaient une fois sur deux, l'installation de Xorg sous Debian, ça plantait. Tu te trouvais devant un prompt et fallait avancer. Puis tu vires ce qui marche pas. Donc j'ai passé de plus en plus de temps devant la ligne de commande. Puis je l'ai raccordé au réseau, et c'était plus confortable de passer par ssh. Voilà comment ça s'est passé.

L'utilisation :

Une grosse majorité de mes productions est réalisée avec l'informatique. Ça m'arrive encore de faire des copiers coller à la main, sinon non. Pas avec la ligne de commande stricto sensu, vu que c'est princialement dans un éditeur de texte, la différence est ténue. Problème de l'éditeur de texte bâtard type emacs, qui existe en dehors et dans l'éditeur. Mais j'édite parfois purement en ligne de commande, comme supprimer la ligne 747 dans un fichier.

Je décide régulièrement de ne pas faire trop peur aux jeux. Genre je vais utiliser oepn office. Je peste parce que àça marche pas, je me dis que je prendrai 3 fois moins de temps. Et je reviens sur le soutil que je connais. J'ai un client mil sur mon serveur, et j'ai pas besoinde dégainer mon laptop et mon mutt.

Mais c'est des situation où je ne bosse pas à fond. Typiquement, il faut que j'écrive une circulaire pour les élèves, j'ai Open Office à disposition, hop. Dans certaines situations, comme quand j'avais la Freebox qui marchait pas et au bout de quatre fois à cliquer sur le bouton redémarrer, j'ai fait un stript en python pour le faire. Du coup j'avais une commande au lieu d'en faire deux.



Dès que t'as un truc un peu répétitif à faire, tu vas le faire en ligne de commande. Je suis même prêt à passer deux heures et demi à écrire un script pour le faire de façon élégante alors que ça prenait moins de temps de le faire à la main.

Juste pour le plaisir d'avoir quelque chose qui fonctionnait.

Y'a une élégance dans la factorisation, en fin pour moi. Si la tâche est répétitive, elle l'est pour toi et pour l'ordinateur aussi. Après cette élégance dépend de tout un chacun. Y'a des gens qui vont trouver cette élégance dans un script Perl d'une ligne. Je vais la trouver dans les outils de base d'Unix, donc ne pas utiliser Perl. Je vais la trouver dans la contrainte aussi.

Ça a un coté improductif, oui et non. Pour de la rentabilité temporelle à court terme, c'est improductif, sur le long terme, ça permet d'entretenir, de développer de compétences. Ne pas le faire c'est se priver de la possibilité de se dire. Le jour où la différence temporelle sera inversée, ça va être la galère.

La ligne de commande n'est pas unique. Tu as différentes interfaces à la ligne de commande et différents interpréteurs. Le premier choix s'opère là. Tu peux décider de configurer d'une manière ou d'une autre.

Déjà tu as un premier stade de configuration.

Ensuite tu as pléthore de configurations avec une partie que tu vas repomper, sur des copains, sur Internet. Tiens je ne connaissais pas cette commande, elle fait quoi, je les fais moi même Y'a des choses où tu vas procéder par essai.

Y'a une partie par mimétisme, y'a une partie par documentation, y'a une partie d'habitude.

C'est vraiment configurer ta ligne de commande pour qu'elle réponde à tes besoins, à tes envies.

Mon fichier de configuration de zsh, mon shell, fait plus de 200 lignes.

T'as besoin d'un élément déclencheur : soit la lecture de la documentation, soit l'observation de la pratique des autres, soit en direct, soit par internet. « Tiens ça manière de faire me plaît je vais essayer. » Sachant que tu vas passer à un moment à un autre par la lecture de la documentation.

C'est une des particularités de l'informatique telle que je la conçois et à laquelle le logiciel libre participe. C'est un outil social comme les autres, et un outil de communication pour favoriser la communication. Tout seul, je ne ferai pas d'informatique.



Initiation.

Ca m'est jamais arrivé de mettre quelqu'un devant. Ça m'est arrivé plus d'une fois d'animer des formations. Des gens qui venaient : "nous on veut apprendre s'en servir", parce qu'on a remarqué que c'était quelque chose d'assez puissant.

Je rappelle souvent cette citation de Moglen qui dit en substance : l'interface graphique, c'est l'interface de l'homme des cavernes, c'est « tu pointes et tu grognes ».

Dans la ligne de commande, tu as un vocabulaire, une grammaire, une syntaxe, c'est ça qui en fait un langage.

Y'a pas d'une part les über geeks qui parlent aux ordinateurs et le reste du monde c'est pas aussi manichéen que ça... En fait ce qu'il faut voir, c'est que la ligne de commande n'est pas adaptée à tous les usages. Par exemple, pour le Web, l'image et la vidéo. Or, ces usages et ces formes de communication, c'est une grosse grosse partie de la communication.

Tes usages, tes habitudes te déforment aussi. En ce qui me concerne, je ne vais pas regarder de vidéos sur Internet, ou assez peu d'images. Ou je vais télécharger l'image dans un fichier temporaire et l'afficher.

Dans une ligne de commande, tu vises aussi à l'efficacité. Y'a rien de plus inefficace que lever la main du clavier pour cliquer, grogner, revenir. J'ai pleinement pris compte de l'efficacité en discutant avec un hémiplégique qui n'avait qu'une main. Il disait que la souris devenait relativement pénible à utiliser. Quand t'as plus qu'une main, tu ne peux pas garder une main sur le clavier.

La souris est concomitante à l'interface graphique. Tu peux chercher lisa2 au PARC de Xerox. Les premières interfaces graphiques sur des CRT utilisaient des pointeurs graphiques. Ça doit être 1973.

La problématique elle est à partir du moment où tu es en train de produire du texte (email, IRC, blog, rapport, code) quand tu es en phase de production t'as besoin d'écrire. Or le plus efficace reste de presser des boutons qui correspondent à des lettres.

On le voit avec d'autres dispositifs d'écriture, qui restent peu pratiques à côté de ça. Produire de la donnée avec la simple utilisation de la souris, c'est très dur.



La vraie différenciation elle est dans quel mode je suis. Ou je produis (même quand je chatte) ou est ce que je suis en train de consommer, de lire ou de propager l'information en cliquant sur « tweet ».

C'est de la production écrite originale.

Ce que fournit la ligne de commande, c'est la meilleure intégration produire / lire / produire / reproduire l'information. Dans un client Twitter en ligne de commande. Et tout se fait avec le clavier.

C'est le côté rassurant du WYSIWYG, c'est l'impression que tu obtiens en temps réel tes modifications. Or un ordinateur aujourd'hui c'est du temps partagé. Entre le moment où t'a cliqué et l'action il y a des dizaines des cycles processeur. Et en ligne de commande, c'est un cycle. Et ça c'est une certitude. Ça suppose une réflexion ultra poussée sur l'outil que tu utilises. Mais est ce qu'on a besoin de réflexion poussée pour utiliser un outil. Après une fois que tu vas être confronté à des cas non standard, du vas devoir te documenter et donc t'interroger. Et tout ça tu vas le faire sans te demander : qu'est ce que ça que percer, et est-ce bien ce qu'il fallait faire ? Est-ce qu'il y avait un autre technique pour avoir le même résultat ?

« tu en vois pas ce qu'elle fait » parce que souvent les gens qui utilisent la ligne de commande ont mené une réflexion sur l'utilisation de l'informatique, je pense.

En fait, l'utilisation de la ligne de commande est très liée à l'administration système parce que c'est l'interface la plus efficace. Pour le coup tu as besoin d'une interface expressive qui te permet d'être efficace. Tu vas taper des commandes, tu veux savoir ce qu'elle va effectuer et avec succès ou pas.

Ce que j'appelle être efficace c'est réaliser des tâches en un minimum de temps et une consommation de ressources minimales. L'idée c'est toujours de ça : au final de passer le moins de temps possible devant les tâches élémentaires soit pour en faire plus que d'habitude, soit dégager du temps pour faire autre chose. Être efficace est faire tout ça sans générer de l'énervement. Si c'est le cas c'est pas efficace.

Les deux courbes de progression :

- tu progresses très vite puis tu stagnes
- tu progresses très lentement puis après inversion de la pente

La ligne de commande fait partie de ces outils qui ont été conçus sans aucune limitation, c'est extensible. Et tu sens que l'interface graphique, tu n'es limité qu'aux sim-



ples entrées vers lesquelles tu peux pointer. C'est comme dans une langue étrangère, tes mots te manquent, mais tu peux y arriver au travers d'une périphrase. C'est pas la façon la plus élégante de le dire, mais tu l'as dit, c'est l'essentiel. Typiquement, c'est que cette action n'a pas été prévue par le développeur.

C'est là aussi une des mesures de l'efficacité, c'est : « est-ce que je vais pouvoir tout faire ? »

Questionnaire établi pour la deuxième salve d'entretiens

Le deuxième questionnaire est plus court, et plus orienté « extérieur à la ligne de commande », que le précédent.

Age

Sexe / Genre

Etudes / Travail

Tu es sous quel système d'exploitation ?

Tu définirais comment ton niveau en informatique ?

Tu utilises l'informatique souvent ? Dans quel contexte ?

C'est quoi pour toi un « expert » en infomatique ?

As tu des infomateurs ou des personnes que tu considères comme « expertes » en la matière dans ta famille ? Parmi tes amis ?

Tu parles d'informatique avec eux ? Dans quel cadre ?

As tu dans ton entourage des gens qui utilisent la ligne de commande ?

Quand tu es face à un terminal, ta réaction ?

A quoi la ligne de commande te fait penser ?

T'a-t-on déjà montré comment l'utiliser ? Comment as tu réagi ?

L'utilises-tu maintenant ? Pourquoi faire, dans quel contexte ?

Si tu l'utilises en présence d'autres personnes, tu t'attends à quel genre de réaction ?



Logs de l'entretien, qui s'est effectué sur IRC

Le pseudo a été remplacé par <utilisateur> pour des raisons de vie privée

18 :36 <utilisateur>25 ans

18 :36 <utilisateur>Homme

18 :37 <utilisateur>réfèrent ressource numérique dans une collectivité

18 :37 <utilisateur>OS : MAc OS X et depuis peu archlinux

18 :38 <utilisateur>et en emploi je rajouterais graphiste/graphiste d'animation et ingénieur qualité

18 :38 <utilisateur>bref

18 :38 <Oriane> intéressant ça : tu pourras me dire pourquoi tu t'es mis à arch à un moment ?

18 :38 <utilisateur>oui je peux le dire

18 :38 <utilisateur>parce que j'ai vu y l'utiliser et j'ai trouver ça cool

18 :39 <utilisateur>mon niveau en info : amateur éclairé

18 :39 <Oriane> :D

18 :39 <Oriane> mais du coup, t'as mis une arch sur ton mac ? /o/

18 :40 <utilisateur>nan, je me suis offert un pc

18 :40 <Oriane> la classe

18 :40 <utilisateur>j'ai cherché à faire un dualboot sur mon mac mais... c'est mission impossible

18 :40 <Oriane> je connais ça :')

18 :41 <utilisateur>et pourtant je me suis pris la tête dessus

18 :41 <utilisateur>mais y'a pas de commande UNIX : vas tu te booter sur ma clé espèce de bougre d'âne de pomme

18 :42 <utilisateur>j'utilise l'informatique très souvent, dans mon métier de graphiste, de réfèrent RN, et d'ingé quali

18 :43 <utilisateur>et aussi pour le plaisir, sur un certain cluster jellyfishien pour filer un coup de main

18 :43 <Oriane> (du coup tu gardes le mac et tu t'en sers toujours ?)

18 :44 <utilisateur>un expert informatique connaît aussi bien l'architecture hardware ET software et est bien sur une star en réseau

18 :44 <utilisateur>et manipule tout en ligne de commande pour faire comme dans



les films

18 :44 <Oriane> « pour faire comme dans les films » ?

18 :44 <utilisateur>(je me sers de mon mac au quotidien, le pc que je me suis offert je m'en sers strictement pour de la bisouille)

18 :45 <Oriane> genre, il fait ça que pour ça :D ?

18 :45 <utilisateur>*bisouille

18 :45 <utilisateur>*bisouille

18 :45 <utilisateur>*bidouille

18 :45 <utilisateur>bordel

18 :45 * Oriane donne un d à utilisateur

18 :45 <utilisateur> :D

18 :45 <Oriane> (le fait d'avoir découvert Arch a changé ton utilisation de Mac ?)

18 :46 <Oriane> (oui je rebondis beaucoup, c'est intéressant ce que tu racontes :3

18 :46 <utilisateur>non bien sûr mais garder des ressources en évitant de pomper pour les interfaces graphiques, je trouve ça très utile

18 :46 <utilisateur>(oui, dans le sens ou je cherche mieux à comprendre comment il marche (le mac))

18 :47 <Oriane> okéé

18 :48 <utilisateur>par exemple, je monitore bcp plus mon mac

18 :48 <utilisateur>je kill les processus useless au fur et à mesure histoire de garder un maximum de stabilité quand je bosse sur des soft gourmands 18 :48 <utilisateur>par exemple

18 :49 <Oriane> ce qui n'est pas forcément un réflexe à la base, oui

18 :49 <utilisateur>il n'y a personne a part des collègues que je considères en expert en info

18 :51 <utilisateur>et quelques amis sur IRC :)

18 :52 <utilisateur>je parle d'informatique avec mes amis et ma famille pour les convaincre et leur apprendre à sécuriser leur surf sur le net, protéger leur disque, leur routeur,...

18 :52 <Oriane> o/

18 :53 <utilisateur>là par exemple mon père se tire dans un pays craignos, je lui ai donc appris à utiliser Tails



18 :53 <utilisateur>à se connecter via un VPN
18 :53 <utilisateur>et à chiffrer son disque
18 :54 <Oriane> du coup à ton tour t'a appris un peu de CLI à quelqu'un :)
18 :54 <utilisateur>il y a peu de gens a part des collegues et des "copains" de tele-comix qui utlise les lignes de commande
18 :54 <utilisateur>absolument :D
18 :55 <Oriane> du coup, t'as appris comment ? via irc/tcx ou via ton boulot ?
18 :55 <utilisateur>irc/tcx
18 :56 <utilisateur>au boulot je suis pas responsable de sécu, je me contente d'aider à formuler un souhait d'application pour les différents services et à en établir les couts
18 :56 <utilisateur>et à les soutenir en commission pour prioriser le tout
18 :56 <Oriane> le déclic s'est fait comment du coup ?
18 :56 <utilisateur>pour la ligne de commande ?
18 :56 <Oriane> oui
18 :57 <utilisateur>en gros une fois x m'a vanné, ça m'a énervé contre moi-même noob que j'étais, j'ai donc pris mon browser préféré et filer sur wikipedia pour comprendre
18 :58 <Oriane> oh
18 :58 <Oriane> donc t'a commencé à apprendre tout seul
18 :58 <utilisateur>oui
18 :58 <utilisateur>et j'apprends toujours tout seul
18 :58 <utilisateur>je suis un gros autodidacte
18 :58 <Oriane> t'as trouvé l'info facilement et tout ou c'était galère ?
18 :59 <utilisateur>hmm j'ai trouvé eds lignes de commandes facilement
19 :00 <Oriane> mais pas le reste :D ?
19 :00 <utilisateur>mais des explications claire sur le fonctionnement global non
19 :00 <utilisateur>j'ai du fouiller
19 :00 <utilisateur>deviner certains trucs
19 :02 <utilisateur>tu as d'autres questions ?
19 :02 <Oriane> sur les forums ou sur irc, y'a des gens qui droppent en disant « j'arrête, c'est trop compliqué pour moi en fait »... c'est pas ton cas !
19 :02 <utilisateur>si, ça l'est, mais je ne supporte pas ne pas comprendre, du coup j'y reviens



19 :02 <Oriane> haa

19 :02 <utilisateur>et en général ça marche mieux

19 :03 <utilisateur> :)

19 :03 <Oriane> donc t'as eu un sursaut d'égo et tu t'y es remis :D

19 :03 <utilisateur>c'est ça

19 :03 <utilisateur>j'ai peu honte mais c'est le cas

19 :03 <Oriane> tu conseillerais à quelqu'un qui droppe de perséverer ?

19 :03 <utilisateur>bien sûr

19 :03 <utilisateur>je pars du principe qu'on a rien sans rien

19 :04 <utilisateur>c'est con de se dire "j'comprends pas", je lui dirais pose des questions

19 :04 <utilisateur>même si elle paraissent bête

19 :04 <Oriane> oui

19 :04 <Oriane> j'imagine que t'as pas hésité à en poser toi aussi

19 :04 <utilisateur>non :)

19 :05 <utilisateur>d'ailleurs j'embête bcp les gens de tcx avec mes questions relous de noob

19 :05 <Oriane> :D

19 :05 <utilisateur>mais je me dis que ça doit flatter le leur, d'égo :)

19 :05 <Oriane> pas faux :D

19 :05 <Oriane> hmm, si, y'a une question qu'est passée à la trappe : la première fois que t'as vu quelqu'un utiliser la cli, ça a fait quoi ?

19 :06 <utilisateur>je me suis dit : ça m'enerve je comprends ce qu'il fait, mais qu'est-ce qu'il fait bordel !

19 :06 <utilisateur>*pas

19 :06 <Oriane> :D

19 :07 <Oriane> et tu l'as assomé de questions :D

19 :07 <utilisateur>oui le pauvre

19 :07 <utilisateur>enfin la première fois je m'en foutais

19 :07 <utilisateur>parce qu'en réalité j'estimai ne pas en avoir besoin

19 :08 <utilisateur>c'est vraiment la réplique de x qui m'a fait bouger mon cul

19 :08 <utilisateur>mais tu lui dis pas hein, sa tête va encore gonfler



19 :08 <utilisateur> :)
19 :08 <Orianne> :))
19 :08 <Orianne> secret de la recherche :D
19 :08 <utilisateur> parfait 19 :08 <utilisateur> :)
19 :08 <Orianne> j'anonymise même les noms dans les questionnaires
19 :08 <utilisateur> super
19 :09 <Orianne> à l'inverse, vu que t'as peu de gens qui utilisent la cli autour de toi apparemment, t'as eu des réactions quand ils t'ont vu l'utiliser ?
19 :09 <Orianne> (je crois que c'est la dernière question, youhou)
19 :09 <utilisateur> oui, mon père m'a dit "haaaa mon fils est un hacker)
19 :10 <utilisateur> ce à quoi j'ai dit lol
19 :10 <Orianne> il pensait hacker = hiii je pirate la banque
19 :10 <utilisateur> oui c'est ça :)
19 :10 <Orianne> ou hacker dans le sens noble du terme ?
19 :10 <Orianne> :D
19 :10 <utilisateur> non non premier choix votre honneur
19 :11 <Orianne> il a fini par comprendre que c'était pas le cas visiblement :D
19 :11 <utilisateur> et quand je lui ai montré, il m'a dit : "pffff frimeur"
19 :11 <Orianne> ahah :D
19 :12 <utilisateur> mais j'ai gagné en crédibilité dans mon entourage
19 :12 <Orianne> concernant ton niveau en informatique ?
19 :12 <utilisateur> genre le mec il utilise la cli, c'est un bonhomme de l'info, vaut mieux le croire, dans le doute
19 :13 <Orianne> le pouvoir de persuasion de la cli, c'est beau :')
19 :13 <utilisateur> tu as vu, je voulais pas y croire au départ
19 :13 <utilisateur> des fois je fais juste "ipconfig" et ça suffit :p
19 :13 <utilisateur> ou net config workstation
19 :14 <utilisateur> (s/ windows
19 :14 <utilisateur>)
19 :14 <Orianne> et là tout le monde fait « hoooo » ?
19 :14 <utilisateur> non ils s'en vont de peur d'être mêlés à un truc louche :p
19 :14 <utilisateur> nopn je blague bien sur



- 19 :15 <utilisateur> en général il n'y a pas de réaction particulière
- 19 :16 <utilisateur> ça te va ?
- 19 :16 <Oriane> oui ø/
- 19 :16 <utilisateur> you need moar ?
- 19 :16 <utilisateur> cool
- 19 :16 <Oriane> si t'as envie d'en raconter plus tu peux, hein :D
- 19 :16 <Oriane> mais là j'ai pas mal de matière déjà
- 19 :16 <utilisateur> ben si tu veux poser plus de questions tu peux hein
- 19 :16 <Oriane> hmm
- 19 :16 <Oriane> awais
- 19 :17 <Oriane> t'as déjà lancé un truc dans un terminal, genre sciemment pour faire peur
- 19 :17 <Oriane> ?
- 19 :17 <utilisateur> lol oui une fois, une seule
- 19 :18 <Oriane> ça a marché ? :D
- 19 :18 <utilisateur> j'ai lancé un redoutable "du" pour son défilement vitesse grand V en vert sur fond noir
- 19 :18 <utilisateur> ça a fait son petit effet
- 19 :18 <utilisateur> :p
- 19 :18 <Oriane> la personne en face a fait quoi ?
- 19 :18 <Oriane> « hiii mais tu vas casser mon pécééé ? »
- 19 :19 <utilisateur> un truc dans le genre oui
- 19 :19 <Oriane> (ils sont prévisibles, les gens, cénul)
- 19 :20 <utilisateur> (héhé oui c'est pour ça que c'est drôle qu'une fois)
- 19 :20 <Oriane> tu lui as expliqué ou t'as laissé le doute planer ?
- 19 :20 <utilisateur> j'ai expliqué
- 19 :21 <Oriane> il a dû être déçu
- 19 :21 <utilisateur> un peu mais moi j'avais fait ma blague j'étais content
- 19 :22 <utilisateur> et puis maintenant il fait "top" et "df" régulièrement
- 19 :22 <utilisateur> donc c'est peut être le début de quelque chose
- 19 :23 <Oriane> ø.
- 19 :23 <Oriane> ø/



19 :24 <utilisateur> :)

19 :24 <Oriane> c'ets bien, ça l'a pas dégoûté

19 :24 <utilisateur> non non :)

19 :26 <Oriane> :D

19 :26 <utilisateur> non parce que le truc avec la cli, c'est qu'on comprend assez vite que c'est très utile et très puissant

19 :27 <utilisateur> enfin bref

19 :27 <utilisateur> je décolle

19 :27 <Oriane> oké

19 :27 <utilisateur> a +

19 :27 <Oriane> merci pour tout ø/

19 :27 <utilisateur> mais de rien

19 :28 <utilisateur> a bientôt

19 :28 <Oriane> oui, à bientôt o/



Logs d'une discussion sur IRC à propos de la ligne de commande

Seule la première lettre du pseudo de chaque personne a été conservé, pour des raisons de vie privée

21 :46 <Oriane> je fais un mémoire
21 :46 <Oriane> sur la ligne de commande
21 :46 <M> hein ?
21 :46 <Oriane> donc du coup je suis obligée de tremper dans les interface pour comprendre mon sujet
21 :46 <Oriane> sur la CLI, wep
21 :46 <M> ah formidable
21 :47 <M> un mémoire dans quel domaine ?
21 :47 <Oriane> en communication
21 :47 * Oriane est au Celsa
21 :47 <M> connais pas, mais c'est donc l'aspect interface homme-machine qui t'intéresse ?
21 :48 <Oriane> oui
21 :48 <Oriane> en fait deux choses
21 :48 <Oriane> l'aspect interface homme-machine
21 :48 <Oriane> et tout ce qu'il y a comme imaginaire autour
21 :48 <Oriane> genre
21 :49 <Oriane> d'un côté : tu utilises la CLI, tu es un dangereux hacker
21 :49 <M> ou quelqu'un qui connaît son métier, mébon
21 :49 <Oriane> oui
21 :49 <Oriane> je parle d'imaginaire, hein
21 :49 <Oriane> :D
21 :49 <M> mais oué je vois le concept
21 :49 <M> quoique
21 :50 <M> y'a de très mauvais films ou les hackers utilisent des interfaces 3d pleine d'effets ridicules
21 :50 <Oriane> (lesquels ?)



21 :50 <M> pff, de ce point de vue celui qui m'a le plus arraché les yeux c'était Swordfish jcrois

21 :50 <Oriane> je note :p

21 :50 <M> jurassic park aussi y'a un truc bien niais en 3d

21 :51 <Oriane> ohh je l'ai vu, j'aurai du y penser

21 :51 <M> pis de manière générale toutes les séries avec un crackage de mot de passe ou les lettres sont trouvées les unes après les autres

21 :51 <Oriane> :')

21 :51 <M> et ou ça fait un gros "access denied" avec une tête de mort au milieu

21 :51 <Oriane> owiiii

21 :51 <M> putain j'ai honte

21 :52 <Oriane> c'est ou le shell cryptique avec des fausses commandes ou ça

21 :52 <M> quand on organise des visites de notre labo pour des lycéens ou plus jeunes, souvent on leur fait une pseudo démo

21 :52 <M> ou on crack un hash sur un cluster

21 :52 <Oriane> hu

21 :53 <M> ben j'ai dû y mettre plein d'effets débiles à la demande des gens qui organisent ça

21 :53 <M> genre pendant que l'attaque tourne ça met des mots de passe au hasard qui clignotent

21 :54 <Oriane> ohh

21 :54 <M> pis pour donner l'impression qu'il se passe quelque chose pendant que l'attaque tourne ça fait des dd au hasard sur les machines pour faire clignoter les leds des disques

21 :54 <T> krr krr krr

21 :54 <M> juste pour faire joli quoi

21 :54 <Oriane> (attention M, je note ce que tu racontes, ça m'intéresse)

21 :55 <M> ah te gêne pas, y'a rien de secret :p

21 :55 * Oriane prévient toujours quand elle note quelque chose

21 :55 <M> ouais donc on parlait des CLI

21 :56 <Oriane> ui

21 :56 <M> tu te focalises sur les CLI des systèmes posix, avec la sémantique du shell



et tout ?

21 :56 <Oriane> alors excellente question

21 :56 <M> ou tu regardes aussi les cli des équipements réseau par exemple

21 :56 <M> qui sont construits dans un esprit très différent

21 :56 <Oriane> je ne sais pas encore où m'arrêter, je pense

21 :57 <M> pis après, dans quelle mesure tu vas déborder sur les langages de programmation

21 :58 <Oriane> hmm ?

21 :58 <M> ben c'est difficile de parler de la sémantique du shell sans aborder un peu la prog et les concepts de l'os

21 :59 <M> mais chaipas si c'est pertinent

21 :59 <M> le shell comme un langage de colle pour orchestrer des processus sur un système multitaches

21 :59 <Oriane> ben

21 :59 <Oriane> de toute manière

22 :00 <Oriane> rien que pour moi c'est intéressant :3

22 :00 <Oriane> autant balayer large et resserrer après le sujet

22 :01 <M> pour que ça soit intéressant faudrait que t'aies l'opinion d'un type assez vieux pour se rappeler l'époque d'avant les cli

22 :01 <M> donc avant les teletypes

22 :01 <Oriane> ouais

22 :01 <M> ceux qui ont bossé avec des rubans ou des cartes perforées

22 :01 <M> jme sens jeune tout a coup, yay

22 :01 <Oriane> dans le bouquin de Levy il en parle un peu

22 :02 <Oriane> donc j'ai un récit de cette époque

22 :02 <Oriane> (Steven Levy - Hackers)

22 :03 <M> hm

22 :03 <M> j'ai jamais appris a taper proprement a 10 doigts 22 :03 <M> j'en utilise que 8

22 :03 <M> mais y'a 10 ans jvoulais apprendre

22 :03 <M> et on m'a dit

22 :04 <M> "ça sert a rien, dans 10 ans tout fonctionnera à reconnaissance vocale"



22 :04 <Oriane> rooooh
22 :04 <M> et meme avant ça, jme rappelle gamin
22 :04 <M> "en l'an 2000 t'auras une bagnole ou tu pourras ouvrir les fenetre avec des commandes vocales"
22 :04 <Oriane> c'est beau :')
22 :04 <M> et des voitures volantes
22 :05 <M> et des réacteurs nucléaires portables
22 :05 <M> a la place, ben on a eu facebook et des iphones :/
22 :05 <Oriane> roh roh roh
22 :05 <Oriane> tu viens de me donner une idée
22 :05 <Oriane> j'ai un bouquin à la maison
22 :06 <Oriane> avec toutes les images de comment ils voyaient l'an 2000 dans les années 60
22 :06 <Oriane> je vais mettre la main dessus :>
22 :06 <Oriane> mais oui
22 :06 <Oriane> enfin
22 :06 <Oriane> on a Siri en 2012, 'tention
22 :06 <M> moi j'ai un album d'images qui appartenait a mon père
22 :06 <M> comment ils voyaient l'énergie nucléaire en 1960
22 :06 <M> ehbahputain
22 :06 <Oriane> huuu
22 :07 <M> "ahouais c'est super on irradie un sandwich aux rayons gamma et 10 ans après il a pas pourri"
22 :08 <M> "hey c'est cool on irradie des fleurs et elles ont des couleurs super jolies"
22 :08 <Oriane> hébé
22 :09 <Oriane> c'est super intéressant ceci dit
22 :09 <Oriane> le décalage entre comment les gens voient la technologie / comment on la leur présente
22 :09 <Oriane> et ce qui finit par arriver
22 :09 <M> le titre c'était "l'atome, puissance magique"
22 :09 <L> Putain, je suis tellement d'accord
22 :10 <L> La techno, les gens voit ça comme magique, ou même les médecins, beau-



coup de gens les voient comme des demi dieux j'ai l'impression

22 :10 <L> x)

22 :11 <M> le nucléaire c'est prométhée 2.0

22 :11 <M> a part ça, par rapport a la perception des gens sur la technologie

22 :11 <Oriane> c'est qu'on voit bien dans un truc comme « l'atome puissance magique » que c'est un discours totalement construit

22 :11 <M> j'avais du faire un méta-devoir assez rigolo y'a qqs années pour mon master

22 :12 <Oriane> aw

22 :12 <M> en fait depuis des années on a intégré au cursus quelques crédits de matières "non-techniques"

22 :12 <Oriane> y'a méta dans ta phrase c'est beau :')

22 :12 <M> genre pour apporter un peu une vision humaine et sociale a l'ingénierie bla bla

22 :12 <Oriane> ouais

22 :12 <M> et donc méta-devoir parce que c'était la dernière année avant la restructuration du programme et bref

22 :12 <M> comme sujet j'avais choisi

22 :13 <M> de fouiller dans les archives du programme, et de déterrer plein de vieux devoirs

22 :13 <M> sur l'impact de la science sur la société et tout

22 :13 <M> et comparer leurs prédictions avec ce qui s'est réellement passé

22 :14 <Oriane> rooh

22 :14 <Oriane> trobien

22 :14 <M> et surtout de comparer la vision des étudiants ingénieurs de l'époque avec la perception de la presse

22 :14 <M> et c'était marrant parce que y'avait une grosse asymétrie

22 :17 <Oriane> M : genre t'as un exemple ?

22 :19 <M> jfais de l'archéologie dans mes backups

22 :19 <M> bon c'était focus sur des sujets informatiques

22 :19 <M> mais y'avait entk : la crainte des changements comportementaux face aux jeux vidéos, la violation de la sphère privée



22 :20 <M> l'info-terrorisme, l'extrémisme religieux en ligne

22 :20 <M> meme le bug de l'an 2000

22 :20 <M> l'électronisation totale des devises et les super-hackers qui viennent voler tout ton fric

22 :20 <Oriane> huuu

22 :21 <L> génial

22 :21 <Oriane> roh le bug de l'an 2000

22 :21 <Oriane> en soi

22 :21 <Oriane> c'est un sujet de thèse

22 :22 <Oriane> tellement y'a eu d'agitation autour

22 :25 <M> ouais j'ai des archives de correspondance mais pour trouver le truc lui même j'vais devoir aller ressortir ça sur un autre stockage :/

22 :25 <M> c'était à l'époque ou j'étais encore en dual boot et ça devait être sur la partoche winwin

22 :27 <M> tiens un truc dont jme rappelle bien c'était

22 :27 <M> chaipas si Clipper/Skipjack ça te dit qqch ?

22 :27 <M> mais j'imagine que non

22 :28 <M> c'était un plan de l'administration Clinton pour rendre la cryptographie omniprésente

22 :28 <M> ils avaient reconnu les risques de sécurité et voulaient que toutes les communications soient sécurisées

22 :28 <M> Skipjack c'était l'algorithme développé spécialement dans ce but

22 :28 <M> et Clipper ça devait être une implémentation hardware pas chère

22 :28 <N> ha un époque il étaient intelligents ?

22 :28 <M> mais l'algo a été développé sans le concours du monde académique

22 :29 <M> alors évidemment pour les mauvaises langues c'était un plan diabolique pour imposer une crypto pour laquelle ils devaient forcément avoir une backdoor

22 :31 <M> bon, on s'est un peu éloignés du sujet, désolé :p

22 :31 <Oriane> roh c'pas grave

22 :31 <L> et finalement il s'est passé quoi ? :-p

22 :32 <M> pour Clipper ? ben tout le monde a hurlé au complot et ça a pas eu du



tout de succès

22 :32 <M> pis après que ça aie mouru ils ont publié l'algo et il a été pété très rapidement après

22 :32 <L> ahah

22 :33 <L> les mecs ont du se sentir cons

22 :34 <M> ouais enfin "très rapidement" pour des standards crypto quoi

22 :34 <M> pété == attaque plus rapide que d'essayer toutes les clefs possibles

22 :36 <Oriane> huuu

22 :37 <M> donc dans l'ensemble ils se sont senti moins cons que, par exemple, les auteurs des failles de validation de la wii ou de la ps3, au hasard

22 :39 <M> bon, revenons-en aux CLI archaïques

22 :39 <Oriane> ovoui

22 :40 <M> a l'ère du pré-ascii, il a eu existe des machines qui utilisaient des langages pas basés sur du texte

22 :40 <Oriane> awi

22 :40 <L> sur des cartes à trou ?

22 :40 <M> les machines APL notamment, langage horriblement complexe pour lequel fallait avoir un clavier avec l'alphabet grec et des symboles mathématiques et plein de merdier bizarre

22 :42 <M> ah et sinon récemment jsuis tombé sur un truc ou le type a eu l'idée géniale de donner une sémantique a la couleur

22 :42 <M> pas comme la coloration syntaxique, ou la couleur aide à reconnaître le texte, mais ou vraiment tu dois colorier ton programme sinon il fait pas la bonne chose

22 :42 <M> t'es daltonien ? ben tant pis pour ta gueule, tu sera pas programmeur

22 :43 <M> tiens voila un exemple de a quoi ça ressemble : <http://www.dnd.utwente.nl/~tim/colorforth/Raystm2/mv050314.html>

22 :43 * Oriane a mal aux yeux

22 :45 <Oriane> A la fin de ses 8h de programmation

22 :45 <Oriane> le mec

22 :46 <Oriane> il doit juste faire une sieste pour reposer ses yeux

22 :46 <M> pis la CLI c'est étroitement couplé a la notion de fichier comme une suite



de bytes qui peuvent être du texte

22 :46 <Oriane> ui

22 :46 <M> ce qui est pas le cas des vieux OS pre-unix si jme trompe pas

22 :47 <M> d'ailleurs meme la notion de fichier...

22 :47 <Oriane> c'est lié à la métaphore du bureau, déjà, non ?

22 :47 <M> bon j'ai un peu l'impression de faire le vieux radoteur la quand meme

22 :47 <M> la notion de fichier ?

22 :48 <M> mhhh jcrois pas

22 :48 <Oriane> fichier / dossier

22 :48 <Oriane> hmm

22 :48 <M> sous dos y'avait des fichiers mais y'avait pas de bureau hein

22 :48 <Oriane> ui

22 :48 <M> d'ailleurs c'est stupide cette notion de fichiers dans des dossiers

22 :48 <M> alors que dans le monde physique la terminologie est dans l'autre sens

22 :49 <Oriane> ow

22 :49 <Oriane> j'avais jamais fait gaffe

22 :49 <Oriane> on met des dossiers dans des fichiers ?

22 :50 <M> on met plutot des fiches, mais la fiche elle fait référence a un dossier après, si c'est chez les RG :p

22 :51 <M> mais meme le nom "fichier" c'est une relique obsolète

22 :51 <M> des vieux OS ou un fichier contenait une suite de "fiches" genre comme des lignes dans une DB

22 :52 <M> le fichier comme blob de bytes générique c'est venu avec unix

22 :55 <M> j'ai lu, mais je sais pas si c'est vrai, que si les commandes de base (cd, ls, etc...) sont si courtes c'est une relique de l'époque ou les claviers étaient lourds et lents et ou tu pouvais facilement te fouler un doigt a force de devoir enfoncer les touches comme un bourrin

23 :00 <Oriane> je pensais que c'était uen question de mémoire système

23 :00 <Oriane> mais ça m'étonne pas

23 :01 <Oriane> et je note anyway

23 :01 fichiers c'était des piles

23 :02 <M> par contre ptet bien que le nom "dossier" ça a été introduit par les mac



23 :02 hypercard
23 :02 <M> et qu'avant on appelait ça répertoire
23 :02 <Oriane> rowiiii
23 :02 <M> hypercard... le flash avant l'heure
23 :03 <Oriane> répertoire
23 :03 le bon vieux system 7 qu'a donné windows via plagiat =)
23 :04 <M> bah ils ont pas plagié grand chose honnêtement
23 :05 <M> a part la corbeille
23 :05 je vendait du matos a cette époque, t'avais tout juste un dos 4 couleur que mac était sous sys7 avec 32k couleurs
23 :05 et les fenetres un peu
23 :05 apple Lisa
23 :06 <Oriane> owi
23 :06 <M> ouais enfin ça s'appelle windows quoi
23 :06 <M> des fenêtres y'en avait depuis la version 1...
23 :06 <Oriane> M : ben y'a aussi le fait que ça marche dans le système anglo saxon
23 :06 bah sys7 c'est avant windows
23 :06 <Oriane> mais moins bien chez nous
23 :06 <M> avant windows 95 certes
23 :07 <M> mais system 7 est sorti quoi, vers 1990 ?
23 :07 meme 3.1 2.0 ou 1
23 :07 le final oui
23 :07 87 genre
23 :08 <Oriane> huhu
23 :08 <M> donc oui, ils se sont piqués des innovations parmi
23 :08 <M> mais ils ont pas tout pompé d'un coup, ça serait gros :)
23 :09 <M> mais jme souviens que le coup de la corbeille ça avait fait gueuler les fanboys apple dans les chaumières
23 :09 bah juste les 'windows' hein =)
23 :09 c'est ça qui a tué steve jobs
23 :09 <M> j'en étais presque un a l'époque en plus
23 :09 <M> mais a ctépoque c'était nul les mac pour bricoler



23 :09 apres tu t'es fait acheter
23 :09 <Oriane> :D
23 :10 <M> stu voulais coder, t'avais quoi, hypercard
23 :10 <M> codewarrior qui coutait quelques k
23 :10 t'vais de l'asm cisc aussi
23 :10 <M> 2-3 trucs d'assembleur 68k
23 :10 ah wé le logo jaune et noir
23 :10 <M> et j'avais trouvé une implémentation d'un basic
23 :10 le wawa préhistorique
23 :10 <M> sans documentation autre que des exemples de sources
23 :10 <M> supayr
23 :11 mais le script était bien puissant
23 :11 <M> presque aussi hardcore que d'essayer de comprendre l'assembleur sans doc
23 :11 <M> j'avais chopé powerfantasm depuis un cd de magazine
23 :11 <M> "cool un nouveau langage"
23 :11 <M> 10 minutes plus loin "wtf c'est quoi cette merde incompréhensible" 23 :12
 c'était un poil ardu wé
23 :12 mais applescript ça ramassait bien
23 :13 avec les extensions dans le finder
23 :14 bon ça valait pas visualbasic ... ;P
23 :15 <M> ouais en effet
23 :15 qu'était clairement du script en moins puissant
23 :15 <M> ahlala les vieux souvenirs
23 :15 <Oriane> :)
23 :15 <M> le premier pote avec un graveur de cd
23 :15 2x
23 :15 <M> l'impression d'être un gros voleur quand tu copies un truc qui dans le commerce vaut quelques milliers d'euros
23 :16 ou tu ferais les 3 premières galettes
23 :16 <M> enfin dizaines de milliers de francs
23 :16 <M> ouais les prières devant le buffer pour pas qu'il se vide



23 :16 photoshop sur 20 d7

23 :16 <M> ou la suite office sur 50

23 :16 =)v 23 :16 <M> jfaisais un truc avec mon vieux mac

23 :16 <M> j'avais pas de lecteur de cd

23 :16 <M> alors j'allais chez un pote qui en allait un

23 :17 <M> et pour les cds qui venaient dans les magazines on piochait et on y foutait en archives splittées sur disquettes

23 :17 <M> et évidemment sur ton paquet de 30 disquettes y'en avait toujours une foireuse

23 :17 clair la 28eme

23 :17 <M> et le "noooooooooes" quand ça foirait a la 15e a la décompression et que hop plus qu'a attendre le week-end suivant

23 :18 une install ça se meritait

23 :18 <M> c'était beau

23 :18 <M> les premiers cd avec la rubrique "charme" aussi

23 :18 les pcx

23 :18 <M> t'as 12 ans tu sais pas cque c'est alors t'y archives

23 :18 <M> et l'adulte qui vient regarder il fait une drôle de tête "mais vous savez ce que c'est ça ?"

23 :19 les gif 16 couleurs par 4

23 :19 <M> "ben... non mais ça a l'air charmant, donc..."

23 :19 <M> ouais le truc débile avec les archives de q pseudo-cryptées via minitel

23 :19 <M> ça c'était épique

23 :19 =)

23 :19 <Oriane> :')

23 :20 je ois avoir 1m3 de d7 et autant de cd roms

23 :20 et encore le graveur scsi

23 :21 un dur externe lacie de 2go les bras levés aussi

23 :22 c'était pas green energy a l'époque

23 :22 <Oriane> pas tellement

23 :23 <Oriane> l'industrie informatique en soi est une catastrophe écolo :')

23 :23 <M> ouep



23 :23 <M> mais bon, on y travaille
23 :23 heureusement le pentium 2 en cartouche est arrivé
23 :23 150°C sans charge cpu
23 :24 <M> ben oui, on y travaille
23 :24 <M> google ils atteignent 1.1 de PUE maintenant
23 :24 un ibm bi proc tu chauffait un gymnase en plein air
23 :24 <Oriane> ui
23 :24 <Oriane> :')
23 :24 <M> pis y'a plein de gens qui se mettent au GPU pour le calcul, ça fait gagner une chiée
23 :25 <M> pis x86 c'est fini
23 :25 <M> tout le monde se prépare a faire de l'arm maintenant
23 :25 juste retour des choses
23 :28 tozst... voilà je cherchais le truc de gravage
23 :28 avec le grille pain
23 :28 <M> ué
23 :28 <M> on s'éloigne résolument des CLI :p
23 :28 <M> tain jme rappelle
23 :28 <M> y'a un truc qui m'énervait sur system 7
23 :29 <M> y'avait une combinaison de touches
23 :29 <M> genre ctrl-pomme-power
23 :29 <M> qui ouvrait une espèce de fenêtre texte
23 :29 wi
23 :29 <M> dans laquelle on pouvait taper mais ça faisait rien
23 :29 le debugger =)
23 :29 <M> et j'ai jamais pu découvrir comment ça marchait
23 :29 <M> sans internet, go figure
23 :29 <Oriane> c'est beau :')
23 :30 <M> ça m'énervait ce truc
23 :30 <M> mais d'une force
23 :30 je pense c'était juste un monitoring
23 :30 <M> passé des nuits a taper des trucs au hasard pour essayer de découvrir



comment ça marchait

23 :30 <M> et RIEN

23 :30 post 'bombes'

23 :30 c'était flippant ça aussi

23 :30 ptet' le ST du coup ça

23 :30 <Oriane> ça faisait quoi ?

23 :31 <Oriane> ça explosait ?

23 :31 le mac c'était le finder qui faisait des grimaces

23 :31 nanan j'ai confondu =)

23 :32 <M> ah non c'était cmd-power

23 :33 <M> cmd-ctrl-power c'était le force reboot

23 :33 powze qu'était une pov touche avec un triangle tordu

23 :34 <Oriane> le clavier mac a bougé du coup ?

23 :34 j'ai pas fais gaffe

23 :34 <M> bougé ?

23 :34 <Oriane> ben

23 :35 <Oriane> tu parles comme si la touche elle existait plus /o/

23 :35 <M> ben c'est le cas non ?

23 :35 elle était encore sur les powerbook

23 :35 'fin les ovnis bicolores

23 :36 <M> hm

23 :36 <Oriane> :')

23 :36 <M> j'ai un g4 qui me sert de serveur

23 :36 <M> j'ai encore le clavier d'origine

23 :36 <M> jvais voir

23 :37 <Oriane> j'en sais rien, le dernier mac que j'ai eu entre les mains est à paris

23 :37 <Oriane> et j'ai jamais fait gaffe

23 :37 c'était une touche deportée en haut a droite

23 :37 <M> ouais y'a pas

23 :38 <M> jme demande si ça a changé au moment ou ils ont abandonnÃ© l'adb au profit de l'usb

23 :38 <Oriane> possible



23 :39 les dongles adb
 23 :39 <Oriane> adb c'est quoi ?
 23 :39 le port apple
 23 :39 <M> apple desktop bus... leur truc pour le clavier et la souris quoi 23 :39
 les modem etc
 23 :40 <Oriane> ahhh
 23 :40 broche ronde
 23 :40 <Oriane> ils avaient un port à eux
 23 :40 <Oriane> huhu
 23 :40 ah wé bien a eux même les salauds
 23 :41 <Oriane> :')
 23 :41 meme le réseau passait par ça 23 :42 te fallait un carton de câble et
 une palette de 'doubleurs' pour faire un lan
 23 :42 <Oriane> du apple pur jus
 23 :42 <M> nan, le réseau passait encore par un autre truc
 23 :42 <M> ça ressemblait au adb mais y'avait des pins en plus
 23 :42 adb ça marchait j'en ai encore
 23 :42 ah ptet' oui
 23 :42 <M> par contre le connecteur clavier-souris, c'est très con, c'était un connecteur
 din compatible avec celui pour le svhs ou qqch
 23 :43 avec 2 fleches opposées (le logo)
 23 :43 <M> me rappelle petit avoir essayé de brancher le clavier a la télé en espérant
 que ça allait magiquement la transformer en ordi
 23 :43 innocent
 23 :43 <Oriane> c'est mignon
 23 :45 pae contre niveau jeux te fallait virtualPC et 3 mois de formation
 23 :45 <Oriane> tu te formais pour jouer
 23 :45 apres y'a eu un warcraft mac
 23 :46 autodidacte =)
 23 :46 <Oriane> aw
 23 :46 <Oriane> mais tout de même
 23 :46 <Oriane> ça creuse de fait un gap entre ceux qui font l'effort et les autres



23 :47 nan j(abuse mais fallait optimiser la ram via des extensions bien obscures

23 :47 ah bah y'a les masters et les autres

23 :48 <M> mon meilleur pote avait deux macs chez lui

23 :48 <M> sa mÃšre avait un bureau a domicile

23 :48 <M> jles avais foutus en rÃ©seau pour pouvoir faire warcraft en multi c'Ã©tait cool

23 :48 la gÃ©nÃ©ration 'à vos mac' c'est quand mÃªme autre que celle d'hacker choice
=)

23 :49 <M> pis chaque fois que jpassais l'associÃ© de sa mÃ¨re pÃ©tait un plomb et dÃ©branchait le rÃ©seau parce que Ã§a faisait planter chaipius quoi

23 :49 <M> à cause d'une extension pourrie que j'avais quand meme fini par dÃ©panner, du coup il s'Ã©tait senti un peu con jcrois

23 :49 l'imprimant laser select

23 :49 <M> OUAIS EXACTEMENT !

23 :49 =)

23 :50 la 360 pour ma part

23 :50 t'avais un buffer a la con

23 :53 mais elle marche encore...

23 :53 (tire une larme)

23 :55 'tin je peux refaire un authentique bureau 'graphiste des 90"

23 :56 powerPC 612x





Cette œuvre est mise à disposition sous licence Attribution - Pas d'Utilisation Commerciale - Partage dans les Mêmes Conditions 3.0 France. Pour voir une copie de cette licence, visitez <http://creativecommons.org/licenses/by-nc-sa/3.0/fr/> ou écrivez à Creative Commons, 44 Castro Street, Suite 900, Mountain View, California, 94041, USA.

RÉSUMÉ ET MOTS-CLÉS

Résumé

Comment une interface peut-elle influencer le rapport à soi, aux autres et à l'informatique ?

Ce travail consiste à explorer les raisons qui poussent une certaine frange des utilisateurs d'ordinateurs à les configurer en ligne de commande, et à en étudier l'impact en terme de présentation de soi.

On remarque en effet que la ligne de commande génère autour d'elle une sociabilité assez particulière, nourrie par des imaginaires ambivalents et théâtralisés. C'est un objet sémiotique puissant, pris dans un réseau d'influences divers qui évoque à lui seul ce que l'on a appelé une esthétique de l'efficacité, ce rapport différent que l'on entretient avec l'informatique en adoptant cet outil.

En effet, cette esthétique excède l'objet auquel elle est attachée – la ligne de commande – et vient couvrir des aspects de la vie quotidienne qui n'ont rien à voir avec l'informatique.

Mots-clés

Ligne de commande, informatique, GNU/Linux, shell, terminal, console, présentation de soi, *hacker*, *geek*, *nerd*.

