

# La ligne de commande pour les poètes

Achille Falaise

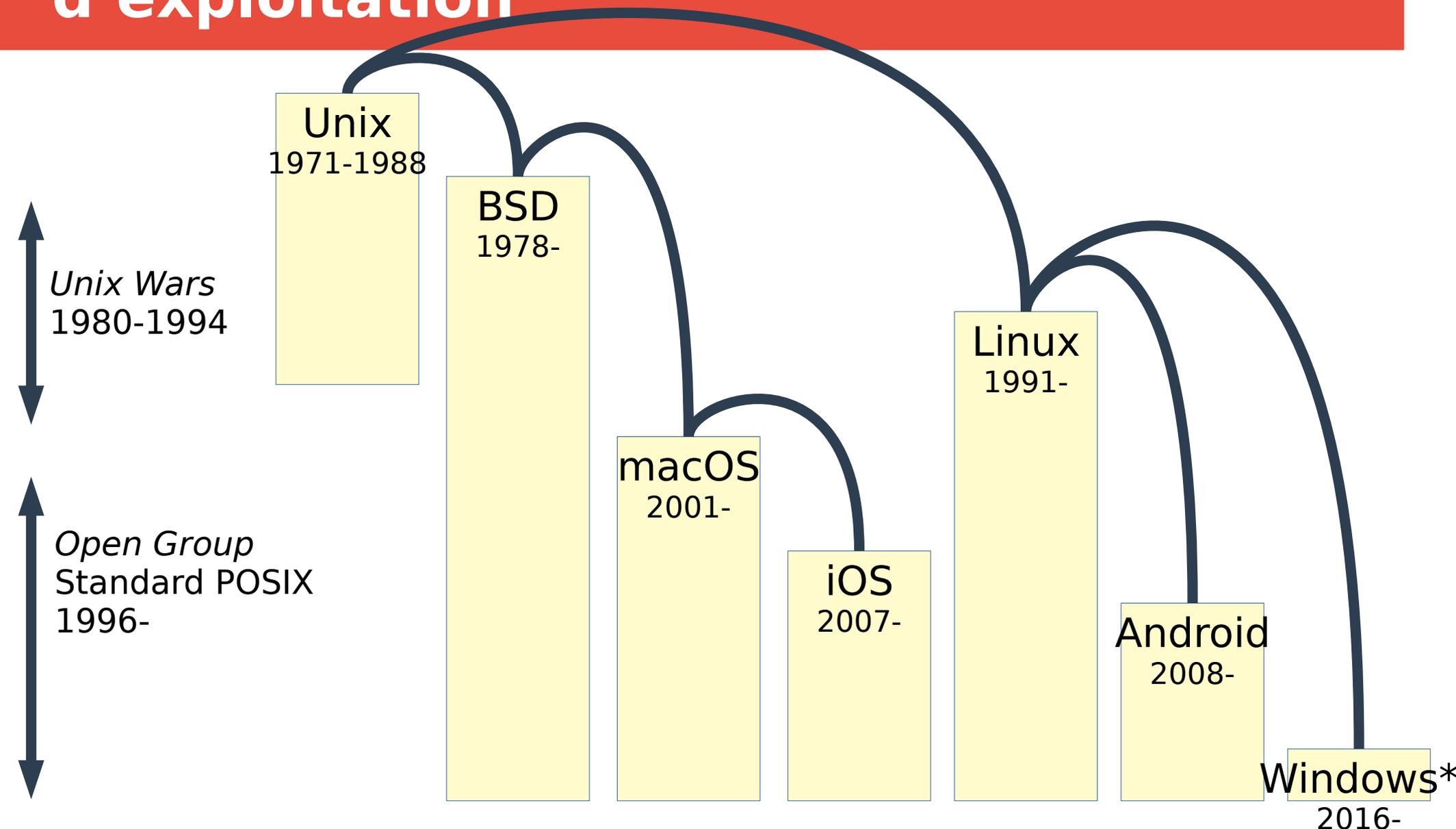
MorDev, mardi 24 juillet 2018

# La ligne de commande pour les poètes

Achille Falaise  
MorDev, mardi 24 juillet 2018

Unix for Poets  
Kenneth Ward Church  
AT&T Bell Laboratories

# Une brève histoire des systèmes d'exploitation

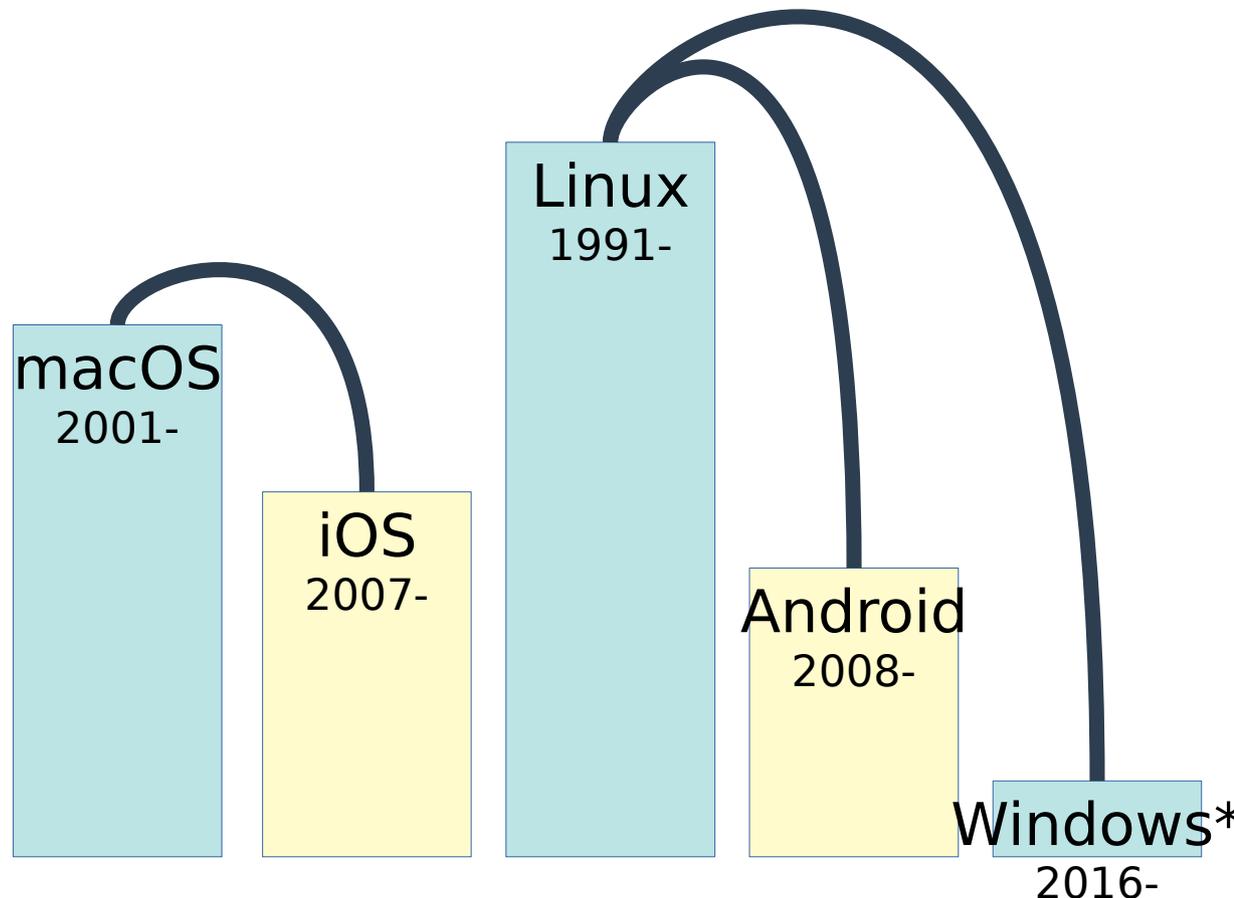


# La « coque logicielle » des systèmes d'exploitation : le *shell*



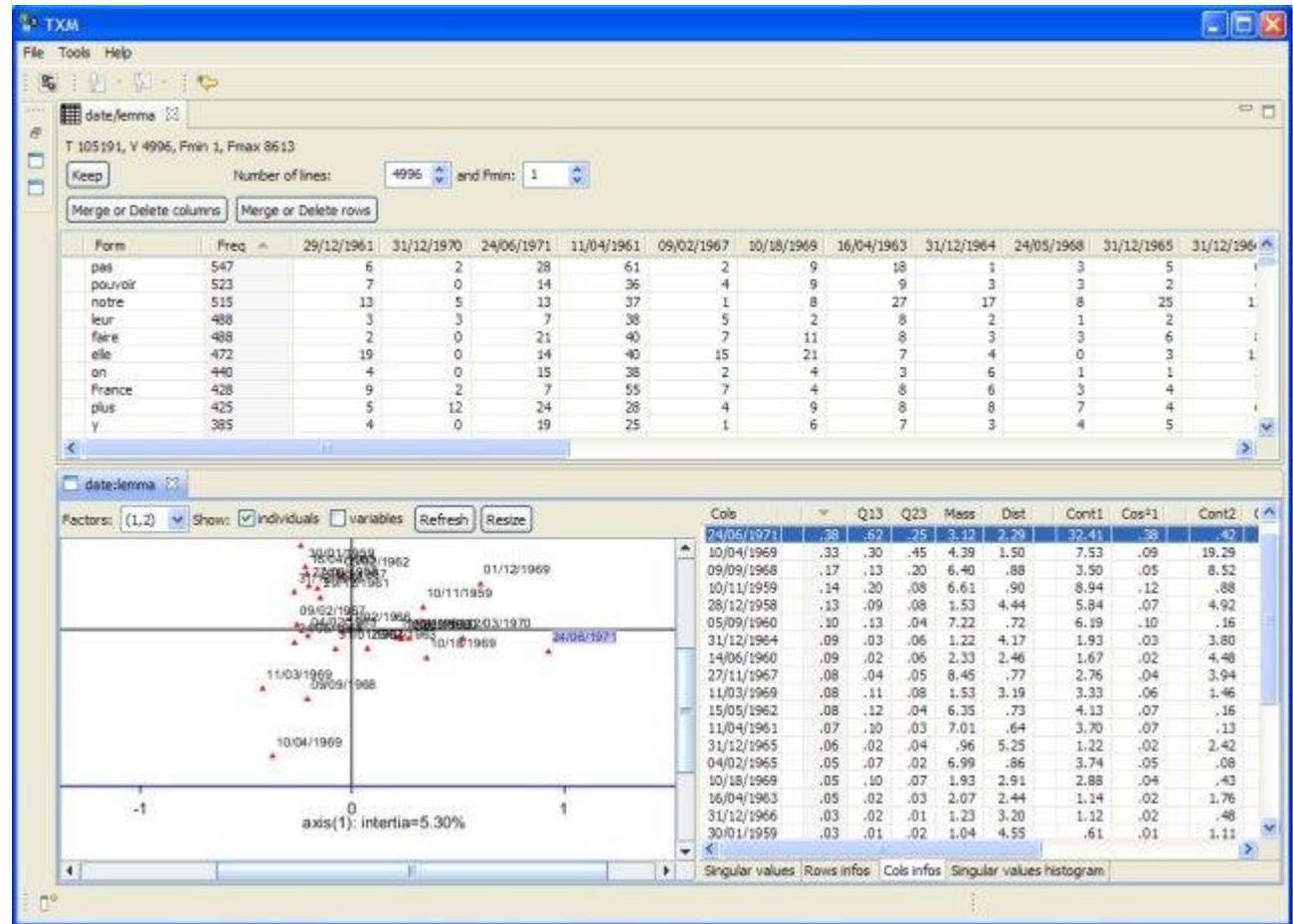
```
kar@kar:~$ pwd
/home/kar
kar@kar:~$ cd /usr/portage/app-shells/bash
kar@kar:~/app-shells/bash$ ls -al
total 136
drwxr-xr-x 3 portage portage 1824 Jul 25 18:06
drwxr-xr-x 33 portage portage 1824 Aug 7 22:59
-rw-r--r-- 1 root root 35880 Jul 25 18:06 ChangeLog
-rw-r--r-- 1 root root 27882 Jul 25 18:06 Manifest
-rw-r--r-- 1 portage portage 4645 Mar 23 21:37 bash-3.1_p17.ebuild
-rw-r--r-- 1 portage portage 5977 Mar 23 21:37 bash-3.2_p09.ebuild
-rw-r--r-- 1 portage portage 6151 Apr 5 14:37 bash-3.2_p48-r1.ebuild
-rw-r--r-- 1 portage portage 5980 Mar 23 21:37 bash-3.2_p48.ebuild
-rw-r--r-- 1 portage portage 5643 Apr 5 14:37 bash-4.0_p10-r1.ebuild
-rw-r--r-- 1 portage portage 6230 Apr 5 14:37 bash-4.0_p10.ebuild
-rw-r--r-- 1 portage portage 5640 Apr 14 05:52 bash-4.0_p17-r1.ebuild
-rw-r--r-- 1 portage portage 5532 Apr 8 18:21 bash-4.0_p17.ebuild
-rw-r--r-- 1 portage portage 5660 May 30 03:35 bash-4.0_p24.ebuild
-rw-r--r-- 1 root root 5660 Jul 25 09:43 bash-4.0_p20.ebuild
drwxr-xr-x 2 portage portage 2848 May 30 03:35 files
-rw-r--r-- 1 portage portage 460 Feb 9 04:35 metadata.xml
kar@kar:~/app-shells/bash$ cat metadata.xml
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE pkgmetadata SYSTEM "http://www.gentoo.org/DTD/metadata.dtd">
<pkgmetadata>
  <herd/base-system/>
  <use>
    <flag name="bashlogger">Log ALL commands typed into bash; should ONLY be
used in restricted environments such as honeypots/</flag>
    <flag name="net">Enable /dev/tcp/host/port redirection/</flag>
    <flag name="plugins">Add support for loading builtins at runtime via
'enable' /</flag>
  </use>
</pkgmetadata>
kar@kar:~/app-shells/bash$ sudo /etc/init.d/bluetooth status
Password:
* status: started
kar@kar:~/app-shells/bash$ ping -q -c 1 en.wikipedia.org
PING rr.esaws.wikimedia.org (91.198.174.2) 56:84 bytes of data:

--- rr.esaws.wikimedia.org ping statistics ---
1 packets transmitted, 1 received, 0% packet loss, time 2ms
rtt min/avg/max/mdev = 49.028/49.028/49.028/0.000 ms
kar@kar:~/app-shells/bash$ grep -l /dev/sda /etc/fstab | cut --fields=3
kar@kar:~/app-shells/bash$ date
Sat Aug 8 02:42:24 HST 2009
kar@kar:~/app-shells/bash$ lsmod
Module      Size  Used by
rmdis_wlan 23424  0
rmdis_host 8696   1 rmdis_wlan
cdc_ether   5672   1 rmdis_host
usbnet     18688   3 rmdis_wlan,rmdis_host,cdc_ether
parport_pc 38424   0
tg3         2388128 28
parport    39648   1 parport_pc
l10n_wdt    12272   0
l2c_core    9398    0
kar@kar:~/app-shells/bash$
```



# CLI (Command Line Interface) VS GUI (Graphical User Interface)

```
root@kali:~# cd /usr/share
root@kali:~/usr/share# cd /usr/share/app-shells/bash
root@kali:~/usr/share/app-shells/bash# ls -al
total 136
drwxr-xr-x 3 portage portage 1624 Jul 25 18:06 .
drwxr-xr-x 33 portage portage 1624 Aug 7 22:39 ..
-rw-r--r-- 1 root root 35880 Jul 25 18:06 ChangeLog
-rw-r--r-- 1 root root 27980 Jul 25 18:06 Manifest
-rw-r--r-- 1 portage portage 4646 Mar 23 21:37 bash-3.1_p17.ebuild
-rw-r--r-- 1 portage portage 5977 Mar 23 21:37 bash-3.2_p39.ebuild
-rw-r--r-- 1 portage portage 6151 Apr 5 14:37 bash-3.2_p48-r1.ebuild
-rw-r--r-- 1 portage portage 5968 Mar 23 21:37 bash-3.2_p48.ebuild
-rw-r--r-- 1 portage portage 5643 Apr 5 14:37 bash-4.0_p18-r1.ebuild
-rw-r--r-- 1 portage portage 6230 Apr 5 14:37 bash-4.0_p18.ebuild
-rw-r--r-- 1 portage portage 5648 Apr 14 05:52 bash-4.0_p17-r1.ebuild
-rw-r--r-- 1 portage portage 5532 Apr 8 18:21 bash-4.0_p17.ebuild
-rw-r--r-- 1 portage portage 5668 May 30 03:35 bash-4.0_p24.ebuild
-rw-r--r-- 1 root root 5568 Jul 25 09:43 bash-4.0_p28.ebuild
-rw-r--r-- 2 portage portage 2348 May 30 03:35 files
-rw-r--r-- 1 portage portage 468 Feb 9 04:35 metadata.xml
root@kali:~/usr/share/app-shells/bash# cat metadata.xml
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE pkgmetadata SYSTEM "http://www.gentoo.org/DTD/pkgmetadata.dtd">
<pkgmetadata>
<chardata-systems></chardata-systems>
<use>
<flag name="bashlogger" Log RLL commands typed into bash; should ONLY be
used in restricted environments such as honeypots//flag>
<flag name="net" Enable /dev/tcp/host/port redirection//flag>
<flag name="plugins" Add support for loading builtins at runtime via
enable' //flag>
</use>
</pkgmetadata>
root@kali:~/usr/share/app-shells/bash# sudo /etc/init.d/bluetooth status
Password:
+ status: started
root@kali:~/usr/share/app-shells/bash# ping -q -c1 en.wikipedia.org
PING rr.esex.wikiimedia.org (93.130.174.2) 56(84) bytes of data:
--- rr.esex.wikiimedia.org ping statistics ---
1 packets transmitted, 1 received, 0% packet loss, time 2ms
rtt min/avg/max/mdev = 49.828/49.828/49.828/0.000 ms
root@kali:~/usr/share/app-shells/bash# grep -l /dev/sda /etc/fstab | out --fields=3
/dev/sda1 /boot
/dev/sda2 none
/dev/sda3 /
root@kali:~/usr/share/app-shells/bash# date
Sat Aug 8 02:42:24 HST 2009
root@kali:~/usr/share/app-shells/bash# lsmod
Module Size Used by
rmdisk_wlan 23424 0
rmdisk_host 1696 1 rmdisk_wlan
cdc_ether 5672 1 rmdisk_host
usbnet 18688 3 rmdisk_wlan,rmdisk_host,cdc_ether
parport_pc 36424 0
fglrx 2386128 20
parport 39648 1 parport_pc
ltd0_wdt 12272 0
ltdc_lbr1 9368 0
root@kali:~/usr/share/app-shells/bash#
```



# CLI (Command Line Interface) VS GUI (Graphical User Interface)

## Intérêt de la ligne de commande (= *shell* sur Unix)

Stable et standard

Précis, documenté

Puissant

Tout logiciel CLI peut fonctionner avec tout logiciel CLI

Mémorisable (dans un fichier texte)

Partageable (par email, sur le Web...)

Automatisable

Permet de créer des programmes (fichiers *.sh*)

# La logithèque Unix

## Une liste de logiciels basiques standardisée

105 logiciels obligatoires

43 logiciels facultatifs

[https://en.wikipedia.org/wiki/List\\_of\\_Unix\\_commands](https://en.wikipedia.org/wiki/List_of_Unix_commands)

+ tous les logiciels que les développeurs jugent utiles

## Des logiciels pas tout à fait standardisés

Implémentation GNU (libre) -> Linux, Windows 10 (mais pas MacOS)



# Récapitulation

	Linux	macOS	Windows (par défaut)	Windows < 10 (Cygwin)	Windows 10 (Sous-système)
Shell Unix	Bash	Bash	-	Bash	Bash
Logithèque Unix	GNU	BSD	-	GNU	GNU

# Anatomie d'une ligne de commande

## Langage formel

Verbe  
(à l'impératif)

Arguments facultatifs

Arguments obligatoires

# Anatomie d'une ligne de commande

## Langage formel

Verbe  
(à l'impératif)

Arguments facultatifs

Arguments obligatoires

ls

ls

Bureau

cd

Bureau

# Anatomie d'une ligne de commande

## Langage formel

Verbe (à l'impératif)	Arguments facultatifs	Arguments obligatoires
ls	169999	Bureau
ls	20180223113832747.pdf 20180316152442504.pdf CR_CNAS_05-10-07.pdf Cytoscape_3_6_1_unix.sh	Bureau
cd	Cytoscape_v3.6.1	Bureau

# Anatomie d'une ligne de commande

## Langage formel

Verbe  
(à l'impératif)

Arguments facultatifs

Arguments obligatoires

ls

ls

Bureau

cd

Bureau

ls

-lsh

Bureau

# Anatomie d'une ligne de commande

## Langage formel

Verbe  
(à l'impératif)

Compléments facultatifs

Compléments obligatoires

```
achille@nyarlathotep:~$ ls -lsh Bureau
total 2,7G
 88K -rw-rw-r-- 1 achille achille 86K mai 16 15:46 169999
304K -rw-rw-r-- 1 achille achille 304K févr. 23 11:47 20180223113832747.pdf
944K -rw-rw-r-- 1 achille achille 944K mars 16 15:33 20180316152442504.pdf
1,7M -rw-rw-r-- 1 achille achille 1,7M mars 27 10:54 CR_CNAS_05-10-07.pdf
299M -rw-rw-r-- 1 achille achille 299M mai 15 17:33 Cytoscape_3_6_1_unix.sh
4,0K drwxrwxr-x 6 achille achille 4,0K juil. 13 11:07 Cytoscape_v3.6.1
104K -rw-rw-r-- 1 achille achille 104K juil. 10 14:29 Devis-100000407785.pdf
104K -rw-rw-r-- 1 achille achille 104K juil. 11 11:24 Devis-100000408102.pdf
156K -rw-rw-r-- 1 achille achille 153K juil. 5 15:30 Devis-D93020939.pdf
```

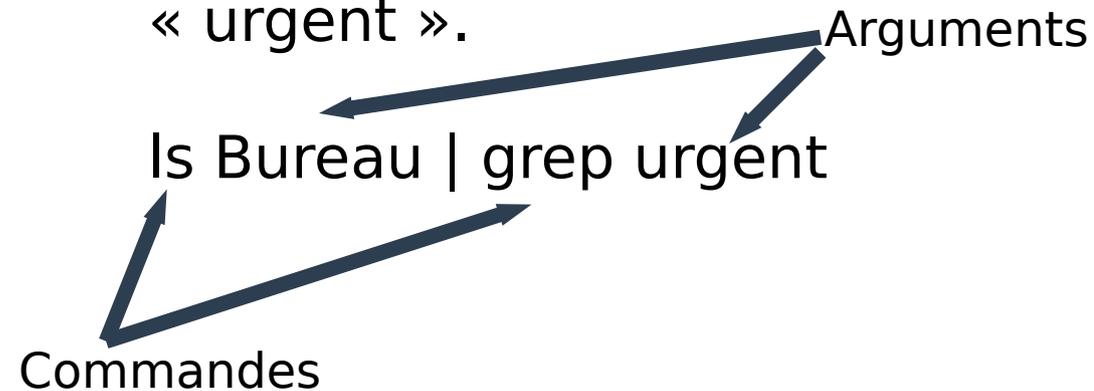
# Anatomie d'une ligne de commande

## Combinaisons de commandes

UneCommande | UneAutreCommande



Exemple : tous les fichiers  
du Bureau qui contiennent  
« urgent ».



# Anatomie d'une ligne de commande

## Envoi vers un fichier

UneCommande > unFichier

UneCommande | UneAutreCommande > unFichier

Exemple : lister tous les fichiers du Bureau qui contiennent « urgent » dans un fichier Àfaire.txt .

```
ls Bureau | grep urgent > Àfaire.txt
```

# Types de commandes

## Actions

**cd** *unDossier* → change de dossier

# Types de commandes

## Affichage (début de chaîne de traitement)

**ls** *unDossier* → affiche le contenu d'un dossier

**ls -lsh** *unDossier* → affichage plus complet

**cat** *unFichier* → affiche le contenu d'un fichier

**cat \*** → affiche le contenu de tous les fichiers du dossier courant

**cat \*.txt** → affiche le contenu de tous les fichiers .txt du dossier courant

**man** *uneCommande* → affiche le manuel d'une commande

**man ls** → affiche le manuel de ls

**man man** → affiche le manuel du manuel

# Types de commandes

## Filtrage

**grep -a unMot** → affiche les lignes qui contiennent un mot

## Tri

**sort** → trie les lignes

## Comptage

**uniq -c** → calcule la fréquence de chaque ligne

**wc** → calcule le nombre de mots

## Remplacement

**sed 's/unTexte/uneAutreTexte/g'** → remplace un texte par un autre

**perl -CS -pe 's/unTexte/uneAutreTexte/g'** → remplace un texte par un autre

# Corpus Littérature avancée

## Préparation

1) Téléchargez et extrayez le corpus Littérature Avancée

<https://www.ortolang.fr/market/corpora/litteracieavancee>

2) Allez dans le dossier où se trouvent les fichiers extraits

cd *nomDeDossier* → va dans le dossier *nomDeDossier*

ls *nomDeDossier*

# Corpus Littérature avancée

## Recherches basiques

1) Affichez tout le corpus (commande **cat**)

- Sur la version simplifiée du corpus :  
cd litteracieavancee  
cat \*
- Sur la version originale du corpus : cat litteracieavancee/1/\*/\*/\*.txt

2) Puis avec un *pipe*...

- Combien de mots dans le corpus ? (**wc**)
- Hypothèse
  - Recherchez toutes les occurrences du mot *hypothèse*  
(**grep -a, grep -a --color**)
  - Combien de lignes contiennent le mot *hypothèse* ? (**wc -l**)

# Corpus Littérature avancée

## Tokenisation

1) Découpage en mots

• **perl -CS -pe** 's/\P{L}+/\n/g'

= « tout ce qui n'est pas une lettre »

= « une nouvelle ligne »

2) Grâce aux tokens

• Combien d'occurrences du mot *hypothèse* ? (**wc -l**)

## Extraction du lexique

1) Triez les mots par ordre alphabétique (**sort**)

2) Calculez la fréquence des mots (**uniq -c**)

3) Triez les mots par fréquence (**sort, sort -n**)

## Préparation

1) Téléchargez et extrayez la base DBPédia

[http://downloads.dbpedia.org/3.7/fr/infobox\\_properties\\_fr.nt.bz2](http://downloads.dbpedia.org/3.7/fr/infobox_properties_fr.nt.bz2)

2) Allez dans le dossier où se trouvent les fichiers extraits

cd *nomDeDossier* → va dans le dossier *nomDeDossier*

ls *nomDeDossier*

## Villes de Lorraine

- 1) Affichez le contenu du fichier
- 2) Filtrez...
  - 1) Sur le mot « Lorraine »
  - 2) Sur la liste des villes de Lorraine
- 3) Nettoyez...
  - Pour n'avoir que les noms de villes (**perl -SC -pe 's/.../.../g'**)
- 4) Sauvegardez...
  - Dans un fichier lorraine.txt

## Préparation

1) Téléchargez et extrayez le corpus de l'Est Républicain (2003) :

<http://www.cnrtl.fr/corpus/estrepublikain/>

2) Allez dans le dossier où se trouvent les fichiers extraits

cd *nomDeDossier* → va dans le dossier *nomDeDossier*

ls *nomDeDossier*

3) Convertissez le corpus en texte :

```
cat Annee2003/*.xml | perl -SC -pe 's/\n/ /g' | perl -SC -pe 's!<teiHeader.*?</teiHeader>!!g' | perl -SC -pe 's/<.*?>//g' > estrep.txt
```

## Recherches sur une liste de mots

- 1) Affichez tout le corpus (commande **cat**)
- 2) Puis avec un *pipe*...
  - Recherchez des concordances sur les villes de Lorraine (**grep -F lorraine.txt**)
  - Quelles sont les villes de Lorraine les plus souvent citées ? (**uniq -c | sort**)

# Outils disponibles en ligne de commande

## Édition de documents tabulaires (Excel, Calc, CSV)

- csvkit

## Édition XML

- Python

## Création de pdf, de graphiques

- LaTeX (documents), graphViz (graphes), imageMagick (images)

## Conversion audio/vidéo

- ffmpeg → conversion automatique

## Téléchargement

wget, curl, httrack, rsync

→ construction de corpus à partir du Web, automatisation de formulaires, archivage...

## Programmation

- Perl, Python, PHP... → programmes personnalisés

## La plupart des outils de TAL

# Exemple 1

```
unzip -p "$LEXICON_ARCHIVE" | # Lire le lexique
perl components/prestoLexicalRules/lexicalRules.pl -r "$LEXICON_RULES" |
csvcut -t -c form,pos,lemma | csvformat -T | # Mettre les colonnes dans
grep -vP "^\\s*#" | # Enlever les commentaires
grep -vP "^\\s*$" | # Enlever les lignes vides
grep -vP "^\\s*form\\tpos\\tlemma" | # Enlever les headers éparpillés dans
grep -vP "^[^\\t]+\\t\\t" | # Enlever les lignes où il n'y a pas de POS
sort -u > "$TMP/lexicon/01_rules.tmp" # Trier et enlever les doublons
```

## Exemple 2

```
cat "$file" |
perl -pe 's/xml:id/xid/g' |
perl -pe 's/\t/ /g' | # Enlever les tabs
perl -pe 's!\[[0-9]+\!\!g' | # Enlever les appels de note
perl -pe 's/\(.*?\)/g' | # Virer les (dgsdgdg ?) et autres (?)
perl -pe 's/\((.*)\)([a-z])/$1$2/g' | # (sédimen)taire -> sédimentaire
perl -pe 's/([a-z])\((.*)\)/$1$2/g' | # pér(iode) -> période
python process.py "$file" | # > out/`basename "$file"`
perl -pe 's@(?<!^<)_@@g' | # Enlever les _trucs_
perl -pe 's@[^<>]+_@@g' | # Enlever les mots pas finis
perl -pe 's!<!\n<!g' |
perl -pe 's!>!\n!g' |
perl -pe 's!^\n$!!g' | # Enlever les lignes vides
perl lib/prestoTokeniser/tokenise.pl -l lexique_morphalou_perceo_oral_modifEIIDA.txt -w 5 |
perl -pe 's/^\s+//' |
perl -pe 's!</?punchme.*?>!!g' |
perl -pe 's/ +$//' | # Enlever les espaces en fin de mot
perl -pe 's!(</(?:p|div)>)\n.\tLOC\n$1!' | # Fin de phrase
perl -pe 's/(<gap .*)\n.\tLOC\n$1/' | # Fin de phrase
perl -pe 's/(<pause .*)\n.\tLOC\n$1/' | # Fin de phrase
perl -pe 's/(<anchor .*)\n.\tLOC\n$1/' | # Fin de phrase
perl -pe 's/(<u .*)\n.\tLOC\n$1/' | # Fin de phrase
perl -pe 's!^\n$!!g' | # Enlever les lignes vides
lib/treetagger/tree-tagger -cap-heuristics -token -lemma -sgml -quiet perceo.par > corpus.ttg
```