

# INTRODUCTION (TRÈS) RAPIDE À LA PROGRAMMATION CSH

IFT6010

Philippe Langlais

[felipe@iro.umontreal.ca](mailto:felipe@iro.umontreal.ca)

RALI

Dept. Informatique et Recherche Opérationnelle

Université de Montréal

- dans un **terminal** :

`ls` liste le contenu du répertoire courant

`cd rep` se déplace dans le répertoire `rep`

`mkdir rep` crée le répertoire `rep` dans le répertoire courant

`cd` se déplace dans le répertoire **home**

`pwd` indique le répertoire courant

- beaucoup de commandes, certaines complexes :

```
1 find . -name "*.cpp" -exec grep -i 'funcZ(' {} \;
```

```
1 echo "bonjour" | rev
```

echo affiche une chaîne

rev renverse le texte qu'on lui donne, ligne à ligne

c1 | c2 l'entrée de c2 est la sortie de c1

```
1 echo "bonjour" | rev >! a
```

c1 >! file la sortie de c1 est **redirigée** dans le fichier file  
(qui est écrasé s'il existe)

c1 >> file la sortie de c1 est **redirigée** dans le fichier file  
(après le contenu existant)

## % man tr

```
TR(1) User Commands
NAME
  tr - translate or delete characters

SYNOPSIS
  tr [OPTION]... SET1 [SET2]

DESCRIPTION
  Translate, squeeze, and/or delete characters from standard input, writing to standard output.

  -c, -C, --complement
      use the complement of SET1

  -d, --delete
      delete characters in SET1, do not translate

  -s, --squeeze-repeats
      replace each input sequence of a repeated character that is listed in SET1 with a single occurrence of that character

  -t, --truncate-set1
      first truncate SET1 to length of SET2

  --help display this help and exit

  --version
      output version information and exit

  SETs are specified as strings of characters. Most represent themselves. Interpreted sequences are:
  \NNN character with octal value NNN (1 to 3 octal digits)
```

```
% head -n 1 zola1.txt
```

Onze heures venaient de sonner à la Bourse, lorsque Saccard entra chez Champeaux, dans la salle blanc et or, dont les deux hautes fenêtres donnent sur la place.

```
% head -n 1 zola1.txt | tr ' ' '\012'
```

Onze  
heures  
venaient  
de  
sonner  
à  
la  
Bourse,  
lorsque ...

```
% head -n 60 zola1.txt | tr ' ' '\012' | grep -i  
ven
```

venaient

**venu**

venait

souvent

**venu**

vendre

venir

- grep recherche une chaîne (ou expression régulière) dans l'**entrée standard**
- toutes les occurrences des chaînes sont affichées sur la **sortie standard**

```
% head -n 200 zola1.txt | tr ' ' '\012' | grep -i  
"^ven" | sort | uniq -c
```

```
1 venaient  
5 venait  
1 vend  
1 vendez,  
2 vendre  
2 venir  
1 vent,  
1 vente  
1 ventre  
2 venu  
1 venue
```

- `sort` tri un fichier (beaucoup d'options disponibles)
- `uniq` élimine les séquences de chaînes identiques (l'option `-c` permet de compter les répétitions)

```
% head -n 200 zola1.txt | tr ' ' '\012' | grep -i  
"^ven" | sort | uniq -c | sort -k1,1nr
```

```
5 venait  
2 vendre  
2 venir  
2 venu  
1 venaient  
1 vend  
1 vendez,  
1 vent,  
1 vente  
1 ventre  
1 venue
```

- l'option `-k1,1nr` donnée à `sort` demande le tri de la première colonne selon un critère numérique (n) et en ordre inverse (r) qui par défaut est croissant

- permet d'introduire des programmes C en shell
- un tutoriel est nécessaire pour maîtriser cette commande
- on peut s'en servir pour afficher une colonne particulière (ici la seconde)

```
% head -n 200 zola1.txt | tr ' ' '\012' | grep -i  
"ven" | sort | uniq -c | sort -k1,1nr |  
awk 'print $2'
```

```
venait  
vendre  
venir  
venu  
venaient  
vend ...
```

- afficher une ligne sous contrainte  
(ici la première colonne doit être supérieure à 2)

```
% head -n 1000 zola1.txt | tr ' ' '\012' | grep  
-i "^ven" | sort | uniq -c | sort -k1,1nr |  
awk '$1 > 2 {print $0}'
```

```
25 venait  
7 venir  
6 venu  
4 Vendôme,  
3 venaient  
3 venant  
3 vendre  
3 vente  
3 ventre
```

## TANNÉS DE TAPER DES COMMANDES À CHAQUE FOIS ?

- Créez un **script**!!!

```
% cat myscript
```

```
1  #!/bin/csh -f
2
3  set in = $1
4
5  cat $in | \
6      awk 'BEGIN {ok=0} \
7          /DEBUT DU FICHER/ {ok = 1} \
8          /FIN DU FICHER/ {ok=0} \
9          {if (ok>10) print $0; else if (ok) ok++}' \
10     | tr ' ' '\012' | tr '[:upper:]' '[:lower:]'
```

- `chmod u+x myscript`
- `./myscript zola1.txt`