

## MEMENTO UNIX

### Se connecter

ssh -l tutoNN -X -i mysshkey hmem.cism.ucl.ac.be

### Afficher les noms de fichiers

ls list  
ls -la list détaillée, long **all**  
ls -ltr list long **time reverse**

### Gérer fichiers et répertoires

mkdir rep make directory  
cd nouveau\_rep change dir  
cd .. répertoire parent  
cd ~ mon homedir  
cd ~bill homedir de bill  
pwd print working dir  
cp fichier\_orig fichier\_dest **copy**  
cp fichier1 fichier2 rép. cp des fichiers dans un rép.  
cp -r rep\_orig rep\_dest recursivement  
ln -s fichier\_orig lien lien symbolique  
mv fichier\_orig fichier\_dest **move** ou renommé  
rm fichier1 fichier2 remove des fichiers ou des liens  
rmdir rep remove dir vide  
rm -rf rep remove dir non vide: **recursif force: danger!**

### Modèles de noms de fichiers

cat \* concate tous les fichiers ordinaires  
cat .\* concate tous les fichiers "cachés"  
cat \*log concate tous les fichiers finissant par log  
ls \*bug\* ls les fichiers avec bug dans leur nom  
ls \*.\*? ls les fichiers finissant par . suivi de 2 caractères

### Gérer le contenu des fichiers

grep souschaine fichier saisir les lignes contenant souschaine  
grep -i souschaine fichier saisir insensible aux maj./min.  
grep -v souschaine fichier saisir tout sauf ce qui contienne souschaine  
grep -r souschaine rep saisir **ré**cusivement dans le répertoire rép.  
sort fichier trier les lignes d'un fichier  
sort -u fichier unique: afficher une seule fois les lignes identiques  
head -<n> fichier affiche les <n> premières lignes  
tail -<n> fichier affiche les <n> dernières lignes  
tail -f fichier affiche les <n> dernières lignes et suit la progression (**f**ollow)

### Droits d'accès aux fichiers

ugo -rwx rwx rwx  
user group other  
chmod u+w fichier user+write +droit d'écrit. du propriétaire  
chmod g-r fichier **g**roup-read, le grp ne peut plus lire)  
chmod o+x fichier **o**ther+execute  
chmod a+rw fichier **a**ll+read+write  
chmod a+rX \* les fichiers rendus executable pour tous  
chmod -R a+rX rep recursive

### Comparer: fichiers, répertoires

diff fichier1 fichier2 comparer 2 fichiers  
diff -r rép1 rép2 comparer 2 répertoires

### Rechercher des fichiers

find . -name "\*log\*" rechercher tous les fichiers dans le répertoire courant (.) avec log dans leur nom  
find rép -name "\*.f90" -exec grep string {} \; tous les fichiers en .f90 dans rép et y saisir "string"

### Rediriger sortie de commande

ls \* > mes\_fichiers rediriger dans un fichier  
ls \*.f >> mes\_fichiers ajouter à un fichier  
cat f1 f2 &> full rediriger output et erreur standard  
cat \*.log | grep error un pipe: rediriger la sortie d'une commande vers l'entrée d'une autre

### Contrôle de tâches

job & lance job en background ( ctrl Z bg)  
bg bakground  
fg foreground  
ps -ef **p**rocess en **e**xecution, **f**ull  
top classement en direct des processus  
kill <pid> tue le process n° indiqué par ps (**p**rocess **i**dentifiant)  
kill -9 <pid> faire tuer pid par le système  
xkill cliquer sur la fenêtre X du programme

### Taille de fichiers et partitions

du -sk rep rep fichier1 fichier2 disk usage **s**ummary **k**ilobytes : l'espace total occupé  
wc fichier word count: nb caractères, mots et lignes

```
df -h .  
df -h
```

**d**isque **f**ilesystem de la partition courante  
Afficher cette info pour toutes les partitions

### Compresser

```
gzip fichier  
gunzip fichier.gz  
ou gzip -d fichier.gz
```

compresser un fichier  
décompresser un fichier.gz

### Manipuler des archives

```
tar -cvf archive.tar rép  
tar -tvf archive.tar  
tar -xvf archive.tar
```

**c**réer une tape archive  
Tester (lister) une archive  
**e**xtraire les fichiers d'une archive

#### Options de tar:

```
c: créer  
t: tester / lister  
x: extraire
```

### Manipuler des archives zip:

```
zip archive.zip <files>  
unzip t archive.zip  
unzip archive.zip
```

créer une archive zip  
**t**ester / lister  
extraire

### Editeur de textes archives

```
nedit fichier &  
xemacs fichier&  
kwrite fichier &  
vi fichier
```

### Soumission de Script SLURM

```
#!/bin/bash
```

```
# Job parameters  
#SBATCH --job-name=demo  
#SBATCH --output=res.txt  
# needed resources  
#SBATCH --ntasks=1  
#SBATCH --time=1:00:00
```

```
# Operations  
echo "Job started at $(date)"  
# Job step  
srun ~/bin/myprog < mydata
```

```
echo "Job end at $(date)"
```

pour le soumettre:

```
sbatch myscript
```