

# TP\_UNIX\_COUFFIN\_2021

October 12, 2021

## 1 Commandes de base pour l'utilisation d'UNIX/Linux

Se référer aux documents sur le système UNIX disponibles sur Moodle pour répondre aux questions posées, ainsi qu'au manuel en ligne (man) pour plus d'informations sur les commandes utilisées.

Sauvegarde le notebook TP\_UNIX\_NOM.ipynb en intégrant votre nom dans le nom du fichier. Une fois le TP terminé, déposer le fichier sur moodle.

### 1.1 I- Comment se repérer

Utilisation des commandes : pwd, ls, ...

1- Depuis la fenêtre de commande utilisée pour le TP (voir ci-avant), afficher la référence absolue de votre répertoire de travail.

Commande : pwd Résultat : /home/cfm3663a

2- Afficher le contenu de votre répertoire de travail avec les fichiers cachés.

Commande : ls -a Résultat : . .bash\_profile .cache .emacs Images Modèles .pki Vidéos .. .bashrc .config .esd\_auth .ipython .mozilla Public .zshrc .bash\_logout Bureau Documents .ICEauthority .local Musique Téléchargements

3- Représentation symbolique des répertoires : quelles différences y a-t-il entre les commandes suivantes ? (a) ls (b) ls . (c) ls .. (d) ls ~ (e) ls /

- (a) Commande : ls Résultats : Bureau Documents Images Modèles Musique Public Téléchargements Vidéos Commentaire : affiche la liste des noms des fichiers et dossiers du repertoire courant
- (b) Commande : ls . Résultats : Bureau Documents Images Modèles Musique Public Téléchargements Vidéos Commentaire : affiche la liste des noms des fichiers et dossiers du repertoire courant
- (c) Commande : ls .. Résultats : bbb2783a brt2086a csm1273a ens@ens.fsi grv3792a lcl1283a mhf8399a njj2646a rjt2832a sst4392a bbb3618a brt2224a csm2020a etudiantfsi ... lck3098a mhb3690a nhr3355a rjm2742a sss2564a brt1540a csl3781a enseignantfsi grv2803a lck3887a mhd2873a njd3348a rjn3125a sss3685a Commentaire : affiche la liste des noms des fichiers et dossiers du repertoire précédent
- (d) Commande : ls ~ Résultats : Bureau Documents Images Modèles Musique Public Téléchargements Vidéos Commentaire : affiche la liste des noms des fichiers et dossiers du repertoire home

- (e) Commande : `ls /` Résultats : `bin dev home lib lost+found mnt proc run srv tmp var winhomes boot etc homelocal lib64 media opt root sbin sys usr winhomep winshare` Commentaire : affiche la liste des noms des fichiers et dossiers de la racine

## 1.2 II- Créer, se déplacer, copier et déplacer des fichiers

Utilisation des commandes : `cd`, `mkdir`, ...

4- Créer le répertoire `TP_UNIX`, se déplacer dans ce nouveau répertoire, et vérifier que c'est bien votre répertoire de travail

Commande : `mkdir TP_UNIX ; cd TP_UNIX ; pwd` Résultats : `/home/cfm3663a/TP_UNIX`  
Vérification : `pwd`

5- Créer le répertoire `Script_Unix`, se déplacer dans ce nouveau répertoire, et vérifier que c'est bien votre répertoire de travail

Commande : `mkdir Script_Unix ; cd Script_Unix ; pwd` Résultats : `/home/cfm3663a/TP_UNIX/Script_Unix`  
Vérification :

6- Récupérer le fichier `boucle.csh` et le fichier `mon_script.csh` sur moodle et faire le nécessaire pour les placer dans le répertoire `Script_Unix`. Ces fichiers sont des scripts Unix (en shell `csh`) qui pourront être exécutés en tapant une commande du type `./nom_du_script.csh`. Vérifier qu'ils sont bien présents dans votre répertoire de travail.

Commande : `mv *.csh ../TP_UNIX/Script_Unix/ ; cd ; cd TP_UNIX/Script_Unix/ ; ls` Résultats : `boucle.csh mon_script.csh` Commentaire : Quand je suis dans le fichier `Script_Unix`, les 2 fichiers sont présents

7- En utilisant la bonne option, lister le contenu du répertoire d'accueil et de ses sous-répertoires. Comment sont organisés les différents répertoires, sous-répertoires et fichiers créés précédemment ?

Commande : `ls -R` Résultats : `.: Bureau Documents Images Modèles Musique Public Téléchargements TP_UNIX Vidéos`

`./Bureau:`

`./Documents:`

`./Images:`

`./Modèles:`

`./Musique:`

`./Public:`

`./Téléchargements: TP_UNIX_COUFFIN_2021.ipynb`

`./TP_UNIX: Script_Unix`

`./TP_UNIX/Script_Unix: boucle.csh mon_script.csh`

`./Vidéos:`

Commentaire : Ils sont organisés comme un arbre.

### 1.3 III- Droits d'accès à un fichier

Utilisation des commandes : cat, cd, ls, chmod, pwd, ...

8- Vérifier que vous êtes bien dans le répertoire Script\_Unix.

Commande : pwd Résultats : /home/cfm3663a/TP\_UNIX/Script\_Unix

9- Afficher le contenu des fichiers .csh qui se trouvent dans ce répertoire, directement depuis le shell, sans passer par un éditeur de texte.

Commande : cat mon\_script.csh Résultats : `#!/bin/csh echo "execution du script en cours" set nb=0 while($nb < 100) echo "....." @ nb = $nb + 1 # attention à bien respecter les espacements end echo "execution du script terminée" #attention de bien revenir à la ligne après`

Commande : cat boucle.csh Résultats : `#!/bin/csh while(1) end #fin de script`

10- Afficher, en utilisant les bonnes options, les informations (format long) correspondant au contenu complet (avec fichiers cachés) du répertoire Script\_Unix.

```
[cfm3663a@u3-204-03l Script_Unix]$ ls -la total 16 drwxr-xr-x 2 cfm3663a utilisateurs du domaine 4096 7 oct. 10:41 . drwxr-xr-x 3 cfm3663a utilisateurs du domaine 4096 7 oct. 10:34 .. -rw-r--r-- 1 cfm3663a utilisateurs du domaine 41 7 oct. 10:37 boucle.csh -rw-r--r-- 1 cfm3663a utilisateurs du domaine 261 7 oct. 10:37 mon_script.csh [cfm3663a@u3-204-03l Script_Unix]$ pwd
```

11- Quels sont les droits d'accès du fichier mon\_script.csh ?

```
-rw-r--r-- 1 cfm3663a utilisateurs du domaine 261 7 oct. 10:37 mon_script.csh
```

C'est cette ligne la venant de la question d'avant nous montre que le propriétaire a les droit d'écriture et de lecture, le groupe a les droit de lecture tout comme les autres (-rw-r--r--).

12- Exécuter la commande chmod 600 ./mon\_script.csh puis refaire la commande de la question 10. Que constatez-vous ?

```
[cfm3663a@u3-204-03l Script_Unix]$ chmod 600 ./mon_script.csh [cfm3663a@u3-204-03l Script_Unix]$ ls -la total 16 drwxr-xr-x 2 cfm3663a utilisateurs du domaine 4096 7 oct. 10:41 . drwxr-xr-x 3 cfm3663a utilisateurs du domaine 4096 7 oct. 10:34 .. -rw-r--r-- 1 cfm3663a utilisateurs du domaine 41 7 oct. 10:37 boucle.csh -rw----- 1 cfm3663a utilisateurs du domaine 261 7 oct. 10:37 mon_script.csh
```

commentaire : les droits du groupe et des "autres" ont changé. Ils n'ont plus aucun droit sur ce fichier.

13- Exécuter la commande ./mon\_script.csh Que se passe-t-il ?

```
[cfm3663a@u3-204-03l Script_Unix]$ ./mon_script.csh bash: ./mon_script.csh: Permission denied
```

COMMENTAIRE : Je n'ai pas les droit d'exécution

14- Que faut-il faire pour que le script puisse s'exécuter ? Modifier les droits d'accès du fichier en conséquence et vérifier que les droits ont bien été modifiés.

```
[cfm3663a@u3-204-03l Script_Unix]$ chmod u+x mon_script.csh [cfm3663a@u3-204-03l Script_Unix]$ ls -la total 16 drwxr-xr-x 2 cfm3663a utilisateurs du domaine 4096 7 oct. 10:41 . drwxr-xr-x 3 cfm3663a utilisateurs du domaine 4096 7 oct. 10:34 .. -rw-r--r-- 1 cfm3663a
```

utilisateurs du domaine 41 7 oct. 10:37 boucle.csh -rwx—— 1 cfm3663a utilisateurs du domaine 261 7 oct. 10:37 mon\_script.csh

15- Refaire la question 13. Que se passe-t-il cette fois ?

```
[cfm3663a@u3-204-03l Script_Unix]$ ./mon_script.csh execution du script en cours .....
.....
..... execution du script
terminee
```

COMMENTAIRE : le fichier s'est exécuté.

16- Exécuter la commande `chmod 777 ./mon_script.csh` et vérifier les droits correspondants. Qu'est-ce qui a été modifié ?

```
[cfm3663a@u3-204-03l Script_Unix]$ chmod 777 ./mon_script.csh [cfm3663a@u3-204-03l
Script_Unix]$ ls -l -a total 16 drwxr-xr-x 2 cfm3663a utilisateurs du domaine 4096 7 oct. 10:41
. drwxr-xr-x 3 cfm3663a utilisateurs du domaine 4096 7 oct. 10:34 .. -rw-r--r- 1 cfm3663a
utilisateurs du domaine 41 7 oct. 10:37 boucle.csh -rwxrwxrwx 1 cfm3663a utilisateurs du domaine
261 7 oct. 10:37 mon_script.csh
```

COMMENTAIRE : Tous les droits d'écriture, d'exécution et de lecture sont donner à tout le monde (owner + group + all).

17- Exécuter la commande `chmod go-rwx ./mon_script.csh` et vérifier les droits correspondants. Quels changements constatez-vous ?

```
[cfm3663a@u3-204-03l Script_Unix]$ chmod go-rwx ./mon_script.csh [cfm3663a@u3-204-03l
Script_Unix]$ ls -a -l total 16 drwxr-xr-x 2 cfm3663a utilisateurs du domaine 4096 7 oct. 10:41
. drwxr-xr-x 3 cfm3663a utilisateurs du domaine 4096 7 oct. 10:34 .. -rw-r--r- 1 cfm3663a utilisateurs
du domaine 41 7 oct. 10:37 boucle.csh -rwx—— 1 cfm3663a utilisateurs du domaine 261 7 oct.
10:37 mon_script.csh
```

COMMENTAIRE : Le group et all n'ont plus leurs droits d'écriture, d'exécution et de lecture mais le owner les garde.

18- Modifier les droits d'accès du fichier `boucle.csh` pour que les membres du groupe (g) et les autres (o) y aient accès en lecture seulement et vérifier les droits avec la commande appropriée.

```
[cfm3663a@u3-204-03l Script_Unix]$ chmod go-r ./boucle.csh [cfm3663a@u3-204-03l
Script_Unix]$ ls -a -l total 16 drwxr-xr-x 2 cfm3663a utilisateurs du domaine 4096 7 oct.
10:41 . drwxr-xr-x 3 cfm3663a utilisateurs du domaine 4096 7 oct. 10:34 .. -rw—— 1 cfm3663a
utilisateurs du domaine 41 7 oct. 10:37 boucle.csh -rwx—— 1 cfm3663a utilisateurs du domaine
261 7 oct. 10:37 mon_script.csh [cfm3663a@u3-204-03l Script_Unix]$ chmod go+r ./boucle.csh
[cfm3663a@u3-204-03l Script_Unix]$ ls -a -l total 16 drwxr-xr-x 2 cfm3663a utilisateurs du
domaine 4096 7 oct. 10:41 . drwxr-xr-x 3 cfm3663a utilisateurs du domaine 4096 7 oct. 10:34
.. -rw-r--r- 1 cfm3663a utilisateurs du domaine 41 7 oct. 10:37 boucle.csh -rwx—— 1 cfm3663a
utilisateurs du domaine 261 7 oct. 10:37 mon_script.csh
```

COMMENTAIRE : tout a fonctionné

## 1.4 IV- Redirection et communication de processus

Utilisation des commandes : `ls`, `wc`, `cat`, ...

Mécanismes de redirection des entrées-sorties standards : <, >, », >&, ...

Mécanismes de communication de processus : |

19- Exécuter la commande cat seule. Que constatez-vous ? Cette commande est un filtre. Qu'est-ce qu'un filtre ?

```
[cfm3663a@u3-204-03l Script_Unix]$ cat
```

COMMENTAIRE : le cat est en pause, il attend un nom de fichier. un filtre attend une entrée et envoie les résultats sur la sortie après les avoir traités.

20- Exécuter la commande suivante puis afficher le contenu du répertoire courant. Que constatez-vous ? A quoi correspond mon\_message ? Que pouvez-vous déduire du rôle du symbole > ?

```
cat > mon_message
echo "bla bla bla ....."
Ctrl D
```

Attention il s'agit de la touche Ctrl de votre clavier !!!!

```
[cfm3663a@u3-204-03l Script_Unix]$ cat > mon_message echo "bla bla bla ....." [cfm3663a@u3-204-03l Script_Unix]$ cat mon_message echo "bla bla bla ....."
```

COMMENTAIRE : On a créé un fichier mon\_message. mon\_message est un nouveau fichier. Le symbole > signifie "envoyé dans".

21- Exécuter la commande cat < mon\_message. Que se passe-t-il ? Que pouvez-vous déduire du rôle du symbole < ?

```
[cfm3663a@u3-204-03l Script_Unix]$ cat < mon_message echo "bla bla bla ....."
```

COMMENTAIRE: Il affiche le message. Le symbole < signifie "redirige vers".

22- Exécuter la commande : cat » mon\_message echo "re bla re bla re bla ....." Ctrl D Attention il s'agit de la touche Ctrl de votre clavier !!!!

```
[cfm3663a@u3-204-03l Script_Unix]$ cat » mon_message echo "re bla re bla re bla ....."
```

et afficher le contenu du fichier mon\_message. Que constatez-vous ? Quel est le résultat ? quel est le rôle du symbole » ?

```
[cfm3663a@u3-204-03l Script_Unix]$ cat < mon_message echo "bla bla bla ....." echo "re bla re bla re bla ....."
```

COMMENTAIRE : La nouvelle ligne rentrée vient s'ajouter à celle d'avant. Le symbole » signifie "ajouter dans".

23- Selon vous que fait la commande suivante : ./mon\_script.csh > resultat\_mon\_script.txt Réfléchir d'abord... Et vérifier ensuite en exécutant la commande, en consultant le contenu du répertoire courant.

HYPOTHESE : je pense que le fichier resultat\_mon\_script.txt va être créé et va suivre l'exécution du fichier mon\_script.csh

```
[cfm3663a@u3-204-03l Script_Unix]$ ./mon_script.csh > resultat_mon_script.txt
```

COMMENTAIRE : Mon hypothèse de départ était bonne.

24- A quoi correspond le fichier resultat\_mon\_script.txt ?

Il correspond à l'exécution du fichier mon\_script.csh.

25- Selon vous que font les commandes suivantes :

```
./new_script.csh
./new_script.csh >& resultat_new_script.txt
```

Réfléchir d'abord... Et vérifier ensuite en exécutant successivement les commandes et en consultant le contenu du répertoire courant.

HYPOTHESE : Il va rediriger la sortie d'erreur du fichier new\_script.csh vers le fichier resultat\_new\_script.txt

```
[cfm3663a@u3-209-08l Script_Unix]$ ./new_script.csh bash: ./new_script.csh: No such file or directory
[cfm3663a@u3-209-08l Script_Unix]$ ./new_script.csh >& resultat_new_script.txt
```

26- A quoi correspond le fichier resultat\_new\_script.txt ?

COPIER-COLLER LE RESULTAT ICI (Commande + résultats + vérification) + COMMENTAIRE

27- Après avoir consulté le manuel en ligne pour vérifier ce que fait la commande wc, exécuter la commande : wc mon\_message A quoi correspond le résultat obtenu ?

```
[cfm3663a@u3-209-08l Script_Unix]$ ls boucle.csh mon_message mon_script.csh resultat_mon_script.txt resultat_new_script.txt
[cfm3663a@u3-209-08l Script_Unix]$ cat resultat_new_script.txt bash: ./new_script.csh: No such file or directory
```

COMMENTAIRE : il redirige vers le fichier new\_script.csh

28- Exécuter les commandes suivantes et en déduire la nature de leurs résultats respectifs : (a) ls -l (b) ls -l | wc (c) ls -l | wc -l (d) ls -l | wc -w (e) ls -l | wc -c (f) cat mon\_message | wc (g) cat mon\_message | wc -lc

En déduire le rôle du symbole |

(a) [cfm3663a@u3-209-08l Script\_Unix]\$ ls -l total 20 -rw-r--r-- 1 cfm3663a utilisateurs du domaine 41 7 oct. 10:37 boucle.csh -rw-r--r-- 1 cfm3663a utilisateurs du domaine 148 7 oct. 11:40 mon\_message -rwx----- 1 cfm3663a utilisateurs du domaine 261 7 oct. 10:37 mon\_script.csh -rw-r--r-- 1 cfm3663a utilisateurs du domaine 2958 7 oct. 11:50 resultat\_mon\_script.txt -rw-r--r-- 1 cfm3663a utilisateurs du domaine 50 12 oct. 12:36 resultat\_new\_script.txt

commentaire : il affiche la repertoire courant version longue

(b) [cfm3663a@u3-209-08l Script\_Unix]\$ ls -l | wc 6 57 425

commentaire : liste le contenu (ligne, caractere et mots) du message qui affiche le contenu du repertoire courant

(c) [cfm3663a@u3-209-08l Script\_Unix]\$ ls -l | wc -l 6

commentaire : affiche le nombre de ligne affiché lorsque que l'on affiche le repertoire courant en version longue.

(d) [cfm3663a@u3-209-08l Script\_Unix]\$ ls -l | wc -w 57

commentaire:affiche le nombre de mots affiché lorsque que l'on affiche le repertoire courant en version longue.

(e) [cfm3663a@u3-209-08l Script\_Unix]\$ ls -l | wc -c 425

commentaire : affiche le nombre de caracteres affiché lorsque que l'on affiche le repertoire courant en version longue.

(f) [cfm3663a@u3-209-08l Script\_Unix]\$ cat mon\_message | wc 9 21 148

commentaire : liste le contenu (ligne, caractere et mots) du fichier mon\_message

(g) [cfm3663a@u3-209-08l Script\_Unix]\$ cat mon\_message | wc -lc 9 148  
commentaire : Affiche le nombre de lignes et de caracteres du fichier mon\_message

## 1.5 V- Gestion des processus

Utilisation des commandes : ps, jobs, kill, CtrlC, CtrlZ, fg, bg, ...

29- Ouvrir un nouvel éditeur de texte avec la commande gedit (ou autre editeur) [cfm3663a@u3-209-08l Script\_Unix]\$ gedit

30- Depuis le terminal, essayer d'afficher le contenu du repertoire courant. Que se passe-t-il ?

[cfm3663a@u3-209-08l Script\_Unix]\$ gedit ls

COMMENTAIRE : il ne se passe rien.

31- Fermer l'éditeur ouvert à la question 29 et refaire la question 30. Qu'en déduisez-vous ?

[cfm3663a@u3-209-08l Script\_Unix]\$ gedit ls //fermeture du document [cfm3663a@u3-209-08l Script\_Unix]\$ ls  
boucle.csh mon\_message mon\_script.csh resultat\_mon\_script.txt resultat\_new\_script.txt

COMMENTAIRE : il affiche le contenu du repertoire commun.

32- Exécuter la commande gedit& puis refaire la question 30. Que constatez-vous ?

[cfm3663a@u3-209-08l Script\_Unix]\$ gedit& [1] 9890 [cfm3663a@u3-209-08l Script\_Unix]\$ ls  
boucle.csh mon\_message mon\_script.csh resultat\_mon\_script.txt resultat\_new\_script.txt

COMMENTAIRE : ça fonctionne

33- Exécuter la commande ps et repérer le PID (identifiant du processus) correspondant à la commande lancée à la question 32. [cfm3663a@u3-209-08l Script\_Unix]\$ ps PID TTY TIME CMD  
8020 pts/1 00:00:00 bash 9890 pts/1 00:00:00 gedit 10055 pts/1 00:00:00 ps

COMMENTAIRE : c'est la 3eme commande

34- Utiliser la commande kill pour supprimer ce processus. Vérifier que le processus a bien été supprimé.

[cfm3663a@u3-209-08l Script\_Unix]\$ kill 9890 [1]+ Terminated gedit [cfm3663a@u3-209-08l Script\_Unix]\$ ps  
PID TTY TIME CMD 8020 pts/1 00:00:00 bash 10354 pts/1 00:00:00 ps

COMMENTAIRE : Le processus n'existe plus.

35- Exécuter le script ./boucle.csh Que se passe-t-il ? Peut-on utiliser la commande kill dans ce cas ? Quelles sont les différentes méthodes permettant d'interrompre, de supprimer ou de suspendre ce processus. Les essayer toutes et relancer au besoin la commande ./boucle.csh

```
[cfm3663a@u3-209-08l Script_Unix]$ ./boucle.csh
```

COMMENTAIRE : il tourne en boucle. Non on ne peut pas utiliser kill car on ne peut pas entrer de nouvelles commandes. Pour l'interrompre on doit utiliser Ctrl C.

```
[cfm3663a@u3-209-08l Script_Unix]$ ./boucle.csh kill kill -9 ^C [cfm3663a@u3-209-08l Script_Unix]$
```

36- Lancer les commandes suivantes puis afficher la liste des processus : (a) head & (b) cat & (c) tail &

```
[cfm3663a@u3-209-08l Script_Unix]$ head & [1] 10986
```

```
[1]+ Stoppé head [cfm3663a@u3-209-08l Script_Unix]$ cat & [2] 11006
```

```
[2]+ Stoppé cat [cfm3663a@u3-209-08l Script_Unix]$ tail & [3] 11083
```

```
[3]+ Stoppé tail [cfm3663a@u3-209-08l Script_Unix]$
```

// J'ai changé d'ordinateur. Je n'ai donc plus les mêmes valeurs de PID pour head, cat et tail.

37- Exécuter ensuite les commandes ci-dessous et comparer les résultats. Consulter le document en ligne sous Moodle et le manuel en ligne pour expliquer le comportement de chacune des commandes : (a) ps (b) ps -ef | grep \$USER (c) jobs (d) jobs -l

```
[cfm3663a@u3-209-07l Script_Unix]$ ps PID TTY TIME CMD 3502 pts/1 00:00:00 bash 4764 pts/1 00:00:00 tail 4799 pts/1 00:00:00 cat 5519 pts/1 00:00:00 head 5557 pts/1 00:00:00 ps [cfm3663a@u3-209-07l Script_Unix]$ ps -ef | grep $USER error: garbage option
```

Usage: ps [options]

Try 'ps -help <simple|list|output|threads|misc|all>' or 'ps -help <s|l|o|t|m|a>' for additional help text.

```
For more details see ps(1). [cfm3663a@u3-209-07l Script_Unix]$ jobs [1] Stoppé tail [2]- Stoppé cat [3]+ Stoppé head [cfm3663a@u3-209-07l Script_Unix]$ jobs -l [1] 4764 Stopped (tty input) tail [2]- 4799 Stopped (tty input) cat [3]+ 5519 Stopped (tty input) head
```

38- Supprimer le processus correspondant à la commande head & en utilisant la commande kill et le PID.

```
[cfm3663a@u3-209-07l Script_Unix]$ kill -9 4626 [1]+ Killed head [cfm3663a@u3-209-07l Script_Unix]$ ps PID TTY TIME CMD 3502 pts/1 00:00:00 bash 4764 pts/1 00:00:00 tail 4799 pts/1 00:00:00 cat 4724 pts/1 00:00:00 ps [cfm3663a@u3-209-07l Script_Unix]$
```

39- Supprimer le processus correspondant à la commande tail & en utilisant la commande kill et le numéro de la tâche. Ce numéro est différent du PID, comment peut-on l'obtenir ?

```
[cfm3663a@u3-209-07l Script_Unix]$ kill %1 [1] Terminated tail [cfm3663a@u3-209-07l Script_Unix]$ ps PID TTY TIME CMD 3502 pts/1 00:00:00 bash 4799 pts/1 00:00:00 cat 5725 pts/1 00:00:00 ps
```

COMMENTAIRE : avec la commande jobs



40- Passer le processus cat & au premier plan, saisir une ligne, observer puis interrompre le processus correspondant.

```
[cfm3663a@u3-209-071 Script_Unix]$ jobs [2]+ Stoppé cat [cfm3663a@u3-209-071 Script_Unix]$ fg %2 cat bonjour bonjour ^C
```

41- Lancer la commande ./boucle.csh en arrière plan, consulter la liste des processus. Mettre le processus correspondant au premier plan et l'interrompre en utilisant la commande appropriée.

```
[cfm3663a@u3-209-071 Script_Unix]$ csh boucle.csh & [1] 7846 [cfm3663a@u3-209-071 Script_Unix]$ jobs [1]+ En cours d'exécution csh boucle.csh & [cfm3663a@u3-209-071 Script_Unix]$ fg %1 csh boucle.csh ^C
```

42- Lancer la commande ./boucle.csh en premier plan, suspendre le processus en utilisant la commande appropriée. Vérifier que le processus est toujours présent. Quel est son PID. Quel est son PPID ? Vérifier également qu'il correspond à une tâche. Quel est le numéro de la tâche ? Supprimer ce processus.

```
[cfm3663a@u3-209-071 Script_Unix]$ csh boucle.csh ^Z [1]+ Stoppé csh boucle.csh [cfm3663a@u3-209-071 Script_Unix]$ ps -f UID PID PPID C STIME TTY TIME CMD cfm3663a 3502 2915 0 17:55 pts/1 00:00:00 bash cfm3663a 8014 3502 26 18:57 pts/1 00:00:02 csh boucle.csh cfm3663a 8053 3502 0 18:57 pts/1 00:00:00 ps -f [cfm3663a@u3-209-071 Script_Unix]$ jobs -l [1]+ 8014 Stopped csh boucle.csh [cfm3663a@u3-209-071 Script_Unix]$ kill %1 [1]- Terminated csh boucle.csh
```

Son PID est 8014, son PPID est 3502 et son numero de tache est 1

43- Exécuter la commande history > mon\_historique. Afficher le contenu du fichier mon\_historique, copier-coller le contenu à la fin du fichier TP1\_VOTRE\_NOM.txt ouvert en début de séance.

```
[cfm3663a@u3-209-071 Script_Unix]$ history > mon_historique [cfm3663a@u3-209-071 Script_Unix]$ cat mon_historique 1 pwd 2 ls -a 3 ls 4 ls . 5 ls .. 6 Bureau Documents Images Modèles Musique Public Téléchargements Vidéos 7 ls ~ 8 cd Musique/ 9 ls ~ 10 ls / 11 cd .. 12 ls / 13 clear 14 mkdir TP_UNIX 15 LS 16 ls 17 cd TP_UNIX/ 18 ls 19 cd .. 20 cd TP_UNIX/ 21 pwd 22 clear 23 mkdir Script_Unix 24 cd Script_Unix/ 25 pwd 26 ./home/cfm3663a/TP_UNIX/Script_Unix 27 clear 28 ./mon_script.csh 29 cd .. 30 cd ~ 31 cd Téléchargements/ 32 ls 33 mv mon_script.csh ../TP_UNIX/Script_Unix/ 34 ls 35 mv boucle.csh ../TP_UNIX/Script_Unix/ 36 ls 37 cd ~ 38 ls 39 pwd 40 cd TP_UNIX/Script_Unix/ 41 ls 42 cd 43 ls -r 44 ls -R 45 clear 46 cd TP_UNIX/Script_Unix/ 47 pwd 48 cat mon_script.csh 49 cat *.csh 50 cat boucle.csh 51 clear 52 ls -l 53 ls -a -l 54 pwd 55 -rw-r-r- 1 cfm3663a utilisateurs du domaine 261 7 oct. 10:37 mon_script.csh 56 clear 57 chmod 600 ./mon_script.csh 58 ls -a -l 59 ./mon_script.csh 60 chmod u+x mon_script.csh 61 ./mon_script.csh 62 ls -l -a 63 clear 64 chmod 777 ./mon_script.csh 65 ls -l -a 66 clear 67 chmod go-rwx ./mon_script.csh 68 ls -a -l 69 chmod go-r ./boucle.csh 70 ls -a -l 71 chmod go+r ./boucle.csh 72 ls -a -l 73 clear 74 cat 75 Elles assurent le traitement d'informations luessur l'entrée et'ecrivent les r'esultats sur la sortie standard ; 76 clear 77 cat > mon_message 78 clear 79 cat > mon_message 80 clear 81 cat > mon_message 82 ls 83 cat mon_message 84 cat < mon_message 85 cat » mon_message 86 cat < mon_message 87 ./mon_script.csh > resultat_mon_script.txt 88 ls 89 cat mon_script.csh 90 cat resultat_mon_script.txt 91 7/10 92 ls 93 jupyter-notebook 94 cd Téléchargements/ 95 jupyter-notebook 96 cd ~ 97 cd .. 98 ./new_script.csh 99 cd ~ 100 ls 101 cd TP_UNIX/ 102 cd Script_Unix/ 103 ./new_script.csh 104 ./new_script.csh >& resultat_new_script.txt 105 ls 106 cat resultat_new_script.txt 107 ls -l (b) ls -l | wc 108 ls -l 109 ls -l | wc 110 ls -l | wc -l 111 ls -l
```

| wc -w 112 ls -l | wc -c 113 cat mon\_message | wc 114 cat mon\_message | wc -lc 115 clear 116  
gedit 117 ls 118 gedit 119 ls 120 gedit& 121 ls 122 ps 123 kill gedit 124 kill -help 125 kill 9890 126  
ps 127 clear 128 ./boucle.csh 129 chmod 777 boucle.csh 130 ./boucle.csh 131 head & 132 cat &  
133 rail & 134 tail & 135 ps 136 ps -ef | grep \$USER  
137 ps -ef | grep \$USER 138 jobs 139 cd TP\_UNIX/ 140 cd Script\_Unix/ 141 ps 142 ps -ef |  
grep \$USER 143 jobs 144 jobs -l 145 clear 146 head & 147 ps 148 kill 4626 149 ps 150 kill -9 4626  
151 ps 152 tail & 153 cat & 154 ps 155 ls -l 156 jobs 157 head & 158 ps 159 ps -ef | grep \$USER  
160 jobs 161 jobs -l 162 kill %1 163 ps 164 kill %3 165 ps 166 clear 167 ./boucle.csh 168 clear 169  
jobs 170 fg %2 171 jobs 172 ps 173 cat & 174 jobs -l 175 fg %1 176 ps 177 clear 178 csh boucle.csh  
& 179 jobs 180 fg %1 181 clear 182 csh boucle.csh & 183 csh boucle.csh 184 ps -f 185 jobs -l 186  
kill %1 187 clear 188 history > mon\_historique