TP_UNIX_GIBERT_Alexis_2021

October 19, 2022

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.

Sauvergarde 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.

Je n'ai pas de nom !@u3-204-11d:~$ pwd /home/gbl3344a

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

Je n'ai pas de nom !@u3-204-11d:/$ ls -a . boot home lib libx32 mnt root snap tmp vmlinuz .. dev initrd.img lib32 lost+found opt run srv usr vmlinuz.old bin etc initrd.img.old lib64 media proc sbin sys var

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) ls : affiche la liste des noms des fichiers et sous-repertoires contenus dans le repertoire courant

Je n'ai pas de nom !@u3-204-11d:/$ ls bin etc initrd.img.old lib64 media proc sbin sys var boot home lib libx32 mnt root snap tmp vmlinuz dev initrd.img lib32 lost+found opt run srv usr vmlinuz.old

ls . : affiche la liste des noms des fichiers et sous-repertoires contenus dans le repertoire courant (meme chose que la commande ls)

Je n'ai pas de nom !@u3-204-11d:/$ ls . bin etc initrd.img.old lib64 media proc sbin sys var boot home lib libx32 mnt root snap tmp vmlinuz dev initrd.img lib32 lost+found opt run srv usr vmlinuz.old

ls .. : affiche la liste des noms des fichiers et sous-repertoires contenus dans le repertoire père du répertoire courant (le répertoire racine n'a pas de père)

Je n'ai pas de nom !@u3-204-11d:/$ ls .. bin etc initrd.img.old lib64 media proc sbin sys var boot home lib libx32 mnt root snap tmp vmlinuz dev initrd.img lib32 lost+found opt run srv usr vmlinuz.old

ls ~ : affiche la liste des noms des fichiers et sous-repertoires contenus dans le repertoire d'acceuil

Je n'ai pas de nom !@u3-204-11d:/$ ls ~ Desktop Downloads

ls / : affiche la liste des noms des fichiers et sous-repertoires contenus dans le repertoire racine

Je n'ai pas de nom !@u3-204-11d:/$ ls / bin etc initrd.img.old lib64 media proc sbin sys var boot home lib libx32 mnt root snap tmp vmlinuz dev initrd.img lib32 lost+found opt run srv usr vmlinuz.old

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

Utilisation des commandes : cd, mkdir, …cd

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

Je n'ai pas de nom !@u3-204-11d:~$ mkdir TP_UNIX Je n'ai pas de nom !@u3-204-11d:~$ ls Desktop Downloads TP_UNIX Je n'ai pas de nom !@u3-204-11d:~$ cd TP_UNIX/ Je n'ai pas de nom !@u3-204-11d:~/TP_UNIX$ Je n'ai pas de nom !@u3-204-11d:~/TP_UNIX$ pwd /home/gbl3344a/TP_UNIX

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

Je n'ai pas de nom !@u3-204-11d:~$ mkdir Script_Unix Je n'ai pas de nom !@u3-204-11d:~$ cd Script_Unix/ Je n'ai pas de nom !@u3-204-11d:~/TP_UNIX/Script_Unix$ pwd /home/gbl3344a/TP_UNIX/Script_Unix

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.

Je n'ai pas de nom !@u3-204-11d:~$ cd Downloads/ Je n'ai pas de nom !@u3-204-11d:~/Downloads$ mv boucle (2).csh .. Je n'ai pas de nom !@u3-204-11d:~/Downloads$ cd .. Je n'ai pas de nom !@u3-204-11d:~$ mv boucle (2).csh TP_UNIX/ Je n'ai pas de nom !@u3-204-11d:~$ cd TP_UNIX/ Je n'ai pas de nom !@u3-204-11d:~/TP_UNIX$ mv boucle (2).csh Script_Unix/ Je n'ai pas de nom !@u3-204-11d:~/TP_UNIX$ cd Script_Unix/ Je n'ai pas de nom !@u3-204-11d:~/TP_UNIX/Script_Unix$ ls 'boucle (2).csh'

Même chose avec 'mon_script (1).csh'

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 ?

Je n'ai pas de nom !@u3-204-11d:~/TP_UNIX/Script_Unix$ cd ~ Je n'ai pas de nom !@u3-204-11d:~$ ls -R .: Desktop Downloads TP_UNIX

./Desktop:

./Downloads: TP_UNIX_GIBERT_Alexis_2021.ipynb unix1.pdf 'TP_UNIX_NOM_ETUDIANT_2021 - Jupyter Notebook.pdf'

./TP_UNIX: Script_Unix

./TP_UNIX/Script_Unix: ‘boucle (2).csh’ ‘mon_script (1).csh’

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.

Je n’ai pas de nom !@u3-204-11d:~/TP_UNIX/Script_Unix$ pwd /home/gbl3344a/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.

Je n’ai pas de nom !@u3-204-11d:~/TP_UNIX/Script_Unix$ cat boucle (2).csh #!/bin/csh while(1) end # fin de script

Je n’ai pas de nom !@u3-204-11d:~/TP_UNIX/Script_Unix$ cat mon_script (1).csh #!/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 terminee” # attention de bien revenir à la ligne après

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

Je n’ai pas de nom !@u3-204-11d:~/TP_UNIX/Script_Unix$ ls -al total 16 drwxr-xr-x 2 gbl3344a 2513 4096 13 oct. 08:49 . drwxr-xr-x 3 gbl3344a 2513 4096 13 oct. 08:36 .. -rw-r–r– 1 gbl3344a 2513 41 13 oct. 08:25 ‘boucle (2).csh’ -rw-r–r– 1 gbl3344a 2513 261 13 oct. 08:49 ‘mon_script (1).csh’

11- Quels sont les droits d’accès du fichier mon_script.csh ?

-rw-r–r– 1 gbl3344a 2513 41 13 oct. 08:25 ‘boucle (2).csh’ leture/écriture pour l’utilisateur lecture pour le groupe lecture pour les autres

-rw-r–r– 1 gbl3344a 2513 261 13 oct. 08:49 ‘mon_script (1).csh’ leture/écriture pour l’utilisateur lecture pour le groupe lecture pour les autres

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

Je n’ai pas de nom !@u3-204-11d:~/TP_UNIX/Script_Unix$ chmod 600 mon_script (1).csh Je n’ai pas de nom !@u3-204-11d:~/TP_UNIX/Script_Unix$ ls -al total 16 drwxr-xr-x 2 gbl3344a 2513 4096 13 oct. 08:49 . drwxr-xr-x 3 gbl3344a 2513 4096 13 oct. 08:36 .. -rw-r–r– 1 gbl3344a 2513 41 13 oct. 08:25 ‘boucle (2).csh’ -rw——- 1 gbl3344a 2513 261 13 oct. 08:49 ‘mon_script (1).csh’

On constate qu’on a supprimé les droit d’accès au groupe et aux autres sur le fichier ‘mon_script (1).csh’

13- Exécuter la commande ./mon_script.csh Que se passe-t-il ?

Je n’ai pas de nom !@u3-204-11d:~/TP_UNIX/Script_Unix$ ./mon_script (1).csh bash: ./mon_script (1).csh: Permission non accordée

Normal car on est pas encore autorisé a executer le fihier (‘x’)

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.

Je n’ai pas de nom !@u3-204-11d:~/TP_UNIX/Script_Unix$ chmod u+x mon_script (1).csh Je n’ai pas de nom !@u3-204-11d:~/TP_UNIX/Script_Unix$ ls -l total 8 -rw-r–r– 1 gbl3344a 2513 41 13 oct. 08:25 ‘boucle (2).csh’ -rwx—— 1 gbl3344a 2513 261 13 oct. 08:49 ‘mon_script (1).csh’

On a maintenant accès en execution au fichier (le ‘x’ à bien été ajouté)

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

Je n’ai pas de nom !@u3-204-11d:~/TP_UNIX/Script_Unix$ ./mon_script (1).csh bash: ./mon_script (1).csh : /bin/csh : mauvais interpréteur: Aucun fichier ou dossier de ce type

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

Je n’ai pas de nom !@u3-204-11d:~/TP_UNIX/Script_Unix$ chmod 777 ./mon_script (1).csh Je n’ai pas de nom !@u3-204-11d:~/TP_UNIX/Script_Unix$ ls -l total 8 -rw-r–r– 1 gbl3344a 2513 41 13 oct. 08:25 ‘boucle (2).csh’ -rwxrwxrwx 1 gbl3344a 2513 261 13 oct. 08:49 ‘mon_script (1).csh’

On remarque que tout les droits (read, write et execute) on été accordés à la fois au groupe mais aussi aux autres. On présume donc que cette commande permet d’affecter les tous les droits a toutes les entités.

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

Je n’ai pas de nom !@u3-204-11d:~/TP_UNIX/Script_Unix$ chmod go-rwx ./mon_script (1).csh Je n’ai pas de nom !@u3-204-11d:~/TP_UNIX/Script_Unix$ ls -l total 8 -rw-r–r– 1 gbl3344a 2513 41 13 oct. 08:25 ‘boucle (2).csh’ -rwx—— 1 gbl3344a 2513 261 13 oct. 08:49 ‘mon_script (1).csh’

On remarque que les droits ‘chmod’ du groupe ‘g’ et des autres ‘o’ on été enlevé ‘-rwx’

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.

Même si c’est dejà le cas on peut changer les droit pour vérifier le changement :

Je n’ai pas de nom !@u3-204-11d:~/TP_UNIX/Script_Unix$ chmod go+rwx ./boucle (2).csh Je n’ai pas de nom !@u3-204-11d:~/TP_UNIX/Script_Unix$ ls -l total 8 -rw-rwxrwx 1 gbl3344a 2513 41 13 oct. 08:25 ‘boucle (2).csh’ -rwx—— 1 gbl3344a 2513 261 13 oct. 08:49 ‘mon_script (1).csh’

On affecte ensuite le seul droit en lecture :

Je n’ai pas de nom !@u3-204-11d:~/TP_UNIX/Script_Unix$ chmod go=r ./boucle (2).csh Je n’ai pas de nom !@u3-204-11d:~/TP_UNIX/Script_Unix$ ls -l total 8 -rw-r–r– 1 gbl3344a 2513 41 13 oct. 08:25 ‘boucle (2).csh’ -rwx—— 1 gbl3344a 2513 261 13 oct. 08:49 ‘mon_script (1).csh’

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 ?

Je n'ai pas de nom !@u3-204-11d:~/TP_UNIX/Script_Unix$ cat ^C On remarque que la commande cat est bloquante. Un filtre est un programme sachant écrire et lire des données par les canaux standards d'entrée et de sortie et qui en modifie ou traite éventuellement le contenu. On comprend donc que tant que l'on lui donne pas des carctères et qu'on ne finit pas la saisie par ^D la commande sera bloquante.

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 !!!!

Je n'ai pas de nom !@u3-204-11d:~/TP_UNIX/Script_Unix$ cat > mon_message echo "blablabla……"

On constate que un fichier "mon message" a été crée dans le répertoire ciblé. Le '>' permet de créer un fichier et plus que ça, la commande cat ouvre le fichier et permet la saie de caractère dans celui-ci.

21- Exécuter la commande cat < mon_message Que se passe-t-il ? Que pouvez-vous déduire du rôle du symbole < ?

Je n'ai pas de nom !@u3-204-11d:~/TP_UNIX/Script_Unix$ cat < mon_message echo "blablabla……"

Cette commande permet d'affichier le contenu d'un fichier, ici mon_message. Par déduction on peut dire que '<' permet d'afficher tandis que '>' permet de saisir.

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 !!!!

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

Je n'ai pas de nom !@u3-204-11d:~/TP_UNIX/Script_Unix$ cat » mon_message echo "re bla re bla re bla ……" Je n'ai pas de nom !@u3-204-11d:~/TP_UNIX/Script_Unix$ cat < mon_message echo "blablabla……" echo "re bla re bla re bla ……"

On constate que le message saisi à été mis a la suite de message. '»' permet de concaténer.

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.

Je présume que la commande a pour but de copier ce que doit afficher le script/programme 'mon_script.csh' dans un nouveau fichier 'resultat_mon_script' au format '.txt' afin de le rendre facilement lisible.

gbl3344a@u3-201-09d:~/TP_UNIX/Script_Unix$ ./mon_script.bash > resultat_mon_script.txt gbl3344a@u3-201-09d:~/TP_UNIX/Script_Unix$ ls 'boucle (2).csh' mon_message mon_script.bash boucle.bash 'mon_script (1).csh' resultat_mon_script.txt

24- A quoi correspond le fichier resultat_mon_script.txt ?

J'ai actuellement pu récuperer les scripts ".bash" fonctionnels, j'ai donc pu obtenir les résulats suivants :

gbl3344a@u3-201-09d:~/TP_UNIX/Script_Unix$ cat < resultat_mon_script.txt execution du script en cours 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 execution du script terminee

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.

Execute le fichier "new_script.csh" et va créer le fichier "resultat_new_script.txt" avant d'y écrire le résulat

gbl3344a@u3-201-01d:~/TP_UNIX/Script_Unix$ ./new_script.csh bash: ./new_script.csh: Aucun fichier ou dossier de ce type gbl3344a@u3-201-01d:~/TP_UNIX/Script_Unix$ ./new_script.csh >& resultat_new_script.txt

26- A quoi correspond le fichier resultat_new_script.txt ?

gbl3344a@u3-201-01d:~/TP_UNIX/Script_Unix$ cat < resultat_new_script.txt bash: ./new_script.csh: Aucun fichier ou dossier de ce type

Le fichier "resultat_new_script.txt" contient l'erreur lors de l'exécuution du fichier "new_script.csh"

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 ?

gbl3344a@u3-201-01d:~/TP_UNIX/Script_Unix$ wc mon_message 2 10 60 mon_message

Le résulat obtenu permet d'indiquer que le fichier mon_message contient "2" lignes, "10" mots et "60" caractères.

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) gbl3344a@u3-201-01d:~/TP_UNIX/Script_Unix$ ls -l total 28 -rw-r–r– 1 gbl3344a 2513 41 13 oct. 08:25 'boucle (2).csh' -rw-r–r– 1 gbl3344a 2513 78 13 oct. 13:41 boucle.bash -rw-r–r– 1 gbl3344a 2513 60 13 oct. 09:39 mon_message -rwx—— 1 gbl3344a 2513 261 13 oct. 08:49 'mon_script (1).csh' -rwxr–r– 1 gbl3344a 2513 259 13 oct. 13:41 mon_script.bash -rw-r–r– 1 gbl3344a 2513 450 13 oct. 13:45 resultat_mon_script.txt -rw-r–r– 1 gbl3344a 2513 60 17 oct. 12:18 resultat_new_script.txt

(b) gbl3344a@u3-201-01d:~/TP_UNIX/Script_Unix$ ls -l | wc 8 67 453

(c) gbl3344a@u3-201-01d:~/TP_UNIX/Script_Unix$ ls -l | wc -l 8

(d) gbl3344a@u3-201-01d:~/TP_UNIX/Script_Unix$ ls -l | wc -w 67

(e) gbl3344a@u3-201-01d:~/TP_UNIX/Script_Unix$ ls -l | wc -c 453

(f) gbl3344a@u3-201-01d:~/TP_UNIX/Script_Unix$ cat mon_message | wc 2 10 60

(g) gbl3344a@u3-201-01d:~/TP_UNIX/Script_Unix$ cat mon_message | wc -lc 2 60

Là où le symbole ";" permet de réaliser plusieurs actions/tâches les unes a la suite des autres le symbole "|" permet de réaliser plusieurs actions/tâches imbriquées les unes dans les autres.

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)

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

gbl3344a@u3-201-01d:~$ gedit (gedit:6278): dbind-WARNING **: 13:04:23.439: AT-SPI: Error retrieving accessibility bus address: org.freedesktop.DBus.Error.ServiceUnknown: The name org.a11y.Bus was not provided by any .service files ls

Rien ne se passe

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

gbl3344a@u3-201-01d:~$ ls Desktop Downloads TP_UNIX gbl3344a@u3-201-01d:~$

On en déduit que lorsque un éditeur de texte est ouvert cela bloque les commande de la console (mais elle sont mémorisés)

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

gbl3344a@u3-201-01d:~$ gedit& [1] 6991 gbl3344a@u3-201-01d:~$ (gedit:6991): dbind-WARNING **: 13:09:15.685: AT-SPI: Error retrieving accessibility bus address: org.freedesktop.DBus.Error.ServiceUnknown: The name org.a11y.Bus was not provided by any .service files ls Desktop Downloads TP_UNIX

On constate que les commande sont exécutés en parallele du fichier ouvert.

33- Exécuter la commande ps et repérer le PID (identifiant du processus) correspondant à la commande lancée à la question 32.

gbl3344a@u3-201-01d:~$ ps PID TTY TIME CMD 5685 pts/1 00:00:00 bash 6991 pts/1 00:00:00 gedit 7221 pts/1 00:00:00 ps

Le PID correspondant à la commande lancée à la question 32 est "6991"

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

gbl3344a@u3-201-01d:~$ kill 6991 gbl3344a@u3-201-01d:~$ ps PID TTY TIME CMD 5685 pts/1 00:00:00 bash 7617 pts/1 00:00:00 ps [1]+ Complété gedit

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

1 1 1 … Que se passe-t-il ? : Le script exécuté est infini (il ne s'arrête jamais)

Peut-on utiliser la commande kill dans ce cas ? : Non

Différentes méthodes permettant d'interrompre : Ctrl+C pour tuer la tâche Ctrl+Z pour suspendre la tâche

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

gbl3344a@u3-201-01d:~/TP_UNIX/Script_Unix$ head & [1] 8839 gbl3344a@u3-201-01d:~/TP_UNIX/Script_Unix$ ps PID TTY TIME CMD 5685 pts/1 00:00:00 bash 8839 pts/1 00:00:00 head 8851 pts/1 00:00:00 ps

[1]+ Stoppé head gbl3344a@u3-201-01d:~/TP_UNIX/Script_Unix$ cat & [2] 8863 gbl3344a@u3-201-01d:~/TP_UNIX/Script_Unix$ ps PID TTY TIME CMD 5685 pts/1 00:00:00 bash 8839 pts/1 00:00:00 head 8863 pts/1 00:00:00 cat 8864 pts/1 00:00:00 ps

[2]+ Stoppé cat gbl3344a@u3-201-01d:~/TP_UNIX/Script_Unix$ tail & [3] 8898 gbl3344a@u3-201-01d:~/TP_UNIX/Script_Unix$ ps PID TTY TIME CMD 5685 pts/1 00:00:00 bash 8839 pts/1 00:00:00 head 8863 pts/1 00:00:00 cat 8898 pts/1 00:00:00 tail 8899 pts/1 00:00:00 ps

[3]+ Stoppé 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

(a) ps : affiche les principales caracteristiques des processus courants appartenant a l'utilisateur qui effectue la commande alexis@alexis-VirtualBox:~$ ps PID TTY TIME CMD 3239 pts/0 00:00:00 bash 5068 pts/0 00:00:00 head 5069 pts/0 00:00:00 cat 5070 pts/0 00:00:00 tail 5076 pts/0 00:00:00 ps

(b) ps –ef | grep $USER : recherche dans l'ensemble de TOUTES les caracteristiques des processus ACTIF Celles
VirtualBox : 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).

(c) jobs : affiche le numero (entre []), l'etat, la commande associee a chaque tache du shell courant ; la tache courante est marquee par +, alexis@alexis-VirtualBox:~$ jobs [1] Stopped head [2]- Stopped cat [3]+ Stopped tail

(d) jobs –l : affiche le numero (entre []), l'etat, la commande associee a chaque tache du shell courant et le PID du processus associe. (Mais cette commande n'as pas l'air de fonctionner sur ma VM) alexis@alexis-VirtualBox:~$ jobs –l bash: jobs: –l: no such job

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

alexis@alexis-VirtualBox:~$ ps PID TTY TIME CMD 3239 pts/0 00:00:00 bash 5068 pts/0 00:00:00 head 5069 pts/0 00:00:00 cat 5070 pts/0 00:00:00 tail 5318 pts/0 00:00:00 ps alexis@alexis-VirtualBox:~$ kill 5068 alexis@alexis-VirtualBox:~$ ps PID TTY TIME CMD 3239 pts/0 00:00:00 bash 5068 pts/0 00:00:00 head 5069 pts/0 00:00:00 cat 5070 pts/0 00:00:00 tail 5318 pts/0 00:00:00 ps

N'a pas marché !

Sur un site j'ai pu trouver les éléments de réponse suivants : " Si un arrêt propre ne fonctionne pas et qu'il faut forcer la fermeture d'un programme, il faut alors envoyer un SIGKILL. Dans ce cas le process n'a pas son mot à dire et c'est l'OS, via INIT qui va se charger du travail de façon violente. Cela peut laisser "traîner" des processus fils orphelins, il faut donc ne l'utiliser que dans les cas désespérés. La commander à taper est : kill -9 "

Pour autant je n'ai pas essayé de lancer la commande.

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 ?

On utilise la commande "jobs" :

alexis@alexis-VirtualBox:~$ jobs [2]- Stopped cat [3]+ Stopped tail alexis@alexis-VirtualBox:~$ kill %3 [3]+ Terminated tail alexis@alexis-VirtualBox:~$ jobs [2]+ Stopped cat

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

alexis@alexis-VirtualBox:~$ fg %2 cat blabla ^Z [1]+ Stopped cat

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.

Je n'ai pas de nom !@u3-205-02d:~/TP_UNIX/Script_Unix$ ./boucle.bash 1 … 1 ^Z Je n'ai pas de nom !@u3-205-02d:~/TP_UNIX/Script_Unix$ bg 1 … 1 Problème : Le processus est inarrétable que ce soit avec la commande ^Z ou avec la commande ^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.

alexis@alexis-VirtualBox:~/TP_UNIX/Script_Unix$ ./boucle.bash 1 … 1 1 ^Z [1]+ Stopped ./boucle.bash alexis@alexis-VirtualBox:~/TP_UNIX/Script_Unix$ ps PID TTY TIME CMD 5691 pts/0 00:00:00 bash 5754 pts/0 00:00:00 boucle.bash 5756 pts/0 00:00:00 ps alexis@alexis-VirtualBox:~/TP_UNIX/Script_Unix$ ps -l F S UID PID PPID C PRI NI ADDR SZ WCHAN TTY TIME CMD 0 S 1000 5691 5680 0 80 0 - 4853 do_wai pts/0 00:00:00 bash 0 T 1000 5754 5691 0 80 0 - 4533 do_sig pts/0 00:00:00 boucle.bas 0 R 1000 5757 5691 0 80 0 - 5013 - pts/0 00:00:00 ps

Le PID de la tâche "boucle.bash" : 5754 Le PPID de la tâche "boucle.bash" : 5691

alexis@alexis-VirtualBox:~/TP_UNIX/Script_Unix$ jobs [1]+ Stopped ./boucle.bash

Le numéro de la tâche “boucle.bash” : 1

alexis@alexis-VirtualBox:~/TP_UNIX/Script_Unix$ kill %1 [1]+ Terminated ./boucle.bash

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.

Je n’ai pas de nom !@u3-205-02d:~/TP_UNIX/Script_Unix$ history > mon_historique

[ ]: