TP1_INF203

Groupe: INF2

Binôme: Shaghayegh HAJMOHAMMADKASHI

Kaiwen ZHENG

[a]: Si on tape une commande inexistante, le système affichera généralement un message d'erreur indiquant que la commande est introuvable ou non reconnue.

Terminal : xyz32 : commande introuvable

[b] : Pour organiser les répertoires principaux des étudiants du DLST sur turing on écrit :

Terminal : cd /home

Et en suite si l'identifient de l'étudiant qu'on veut commence par l'alphabet H on écrit :

Terminal : cd h

Et pour voir le liste des étudiants avec l'identifient commençant par H on écrit :

Terminal : ls

Donc pour accéder au répertoire principal de notre binôme qui est par exemple « hajmohas »,

Chemin relatif : cd ../../h/hajmohas

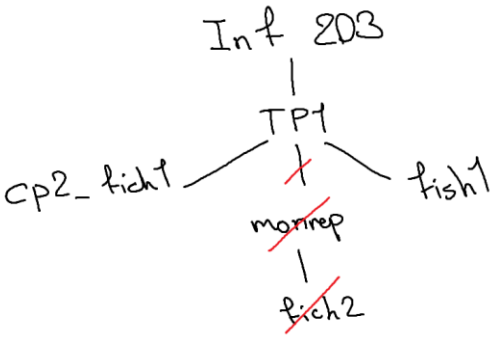
Chemin absolu: cd /home/h/hajmohas

En Linux, un chemin absolu est un chemin complet à partir de la racine du système de fichiers (/) qui spécifie l'emplacement d'un fichier ou d'un répertoire.

Un chemin relatif est un chemin par rapport à l'emplacement actuel du shell utilisateur.

Terminal : mkdir INF203

cp -R /Public/203_INF_Public/TP1 INF203



[c] :

Terminal : cd INF203/TP1

rm a_supprimer

cp fiche1 copie-fiche1

mv copie-fich1 copie2-fich1

mkdir monrep

mv fich2 monrep

[d] :Par exemple pour vérifier le déplacement de fich2 dans monrep

Terminal : cd monrep

ls

Si il affiche fich2 c'est-à-dire on a bien fait le déplacement.

Et pour voir tous on peut aller dans le répertoire TP1 et regarder la liste dans TP1

Terminal : cd ..

Ls

Pour supprimer monrep d'abord on doit supprimer les contenus de monrep parce que la command « rmdir » seulement supprime un répertoire vide donc :

Terminal : cd monrep

rm fich2

rmdir monrep

[e] : Pour supprimer monrep en une seule commande on utilise la commande "rm -r" parce que "rm -r" est une commande Linux utilisée pour supprimer récursivement un fichier ou un répertoire.

"rm" signifie "remove", et l'option "-r" signifie "récursif". Lorsqu'elle est utilisée sur un répertoire, la commande "rm -r" supprime le répertoire ainsi que tout son contenu, y compris ses sous-répertoires et fichiers.

Il est important d'être prudent lors de l'utilisation de cette commande car une fois un fichier ou un répertoire supprimé, il n'est plus possible de le récupérer. Il est donc conseillé de toujours vérifier à deux fois le chemin d'accès avant d'utiliser "rm -r".

[f] : L'option « -l » » dans la commande « ls » sous Linux permet d'afficher la liste des fichiers et répertoires d'un répertoire de manière détaillée, en format long. Il affiche les informations telles que les permissions, le nombre de liens, le propriétaire, le groupe, la taille et les dates de modification des fichiers

Lorsque vous tapez « ls -l » et appuyez sur la touche « Tab » dans un terminal Linux, le shell essayera de compléter la commande ou de suggérer des options ou des arguments possibles que vous pouvez utiliser avec la commande « ls ».

Donc après qu'on appuie une fois sur la touche « Tab » il affiche :

Terminal : ls -l pgcd

Et après qu'on appuie deux fois sur la touche « Tab » il affiche :

pgcd1.c pgcd1.p pgcd2.c pgcd2.p

le plus court préfixe de donneespgcd qu'il faut saisir avant la touche « Tab » pour que ce nom de fichier soit complété sans ambiguïté est le préfixe « don »

Terminal : ls -l donneespgcd

Terminal : cp donneespgcd donquichotte

ls -l don

donneespgcd donquichotte

[g] :Maintenant le plus court préfixe de donneespgcd qu'il faut saisir avant la touche « Tab » pour que ce nom de fichier soit complété sans ambiguïté est le préfixe « donn ».

[h] : la quatrième précédente commande qu'on avait exécutée est ls -l pgcd

Terminal : cp pgcd1.* Prog1

cp *.p Pascal

cp pgcd* Pgcd

rm *pgcd

ls *qui*

donquichotte progquiboucle.c progquiboucle.p

[i]:

Terminal : touch LLL

cat LLL

echo *

donquichotte Etoile LLL Pascal Pgcd pgcd1.c pgcd1.p pgcd2.c pgcd2.p Prog1 progquiboucle.c progquiboucle.p

cp *

cp: la cible 'progquiboucle.p' n'est pas un répertoire

La commande "cp *" dans Linux copie tous les fichiers du répertoire courant dans un autre répertoire. Mais ici on peut pas car 'progquiboucle.p' n'est pas un répertoire.

[j] :

La commande "cp *" copie tous les fichiers présents dans le répertoire courant vers l'emplacement spécifié, soit un autre répertoire ou une destination spécifique. Si aucune destination n'est spécifiée, les fichiers seront copiés dans le répertoire courant.

Terminal :mkdir g u e s

```
cp ille/gui/le
find uil -type f -exec mv -i {} . \;
touch ai s
mv u* ig
mv *e*g aigu
rm -r ???*
find . -mindepth 2 -type f -exec mv {} . \;
find /chemin/du/repertoire/actuel -type d -empty -delete
```

[h] :grep Candide Candide chapitre1.txt

[k] :

La commande "cp *" copie tous les fichiers présents dans le répertoire courant vers l'emplacement spécifié, soit un autre répertoire ou une destination spécifique. Si aucune destination n'est spécifiée, les fichiers seront copiés dans le répertoire courant.

[l] :

Terminal: cd INF203/TP1/scripts

```
cat  sauvegarde.sh
#!/bin/bash
cd $HOME/INF203
cp -r TP1 sauve_TP1
cd sauve_TP1
echo Sauvegarde effectuée dans
pwd
echo il contient
ls -l
```

```
./sauvegarde.sh
./sauvegarde.sh: Permission non accordée
```

Le message d’erreur obtenu est « Permission non accordée »

Terminal : ls -l sauvegarde.sh

-rw-r--r-- 1 hajmohas ima-nogroup 120 26 janv. 15:56 sauvegarde.sh

[m] :

la raison du message d’erreur précédent-rw-r--r-- représente les autorisations d'accès pour le fichier. Les autorisations sont divisées en trois ensembles, chaque ensemble comprenant trois caractères:

Le premier ensemble (rw-) représente les autorisations pour le propriétaire du fichier.

Le deuxième ensemble (r--) représente les autorisations pour le groupe associé au fichier.

Le troisième ensemble (r--) représente les autorisations pour les autres utilisateurs.

Chaque caractère représente une autorisation différente:

r signifie lecture (read)

w signifie écriture (write)

x signifie exécution (execute)

- signifie que l'autorisation est refusée.

Le nombre 1 à côté du premier ensemble d'autorisations (1) représente le nombre de liens hard pour ce fichier.

Les autres éléments de la ligne (par exemple, hajmohas et ima-nogroup) représentent respectivement le propriétaire et le groupe du fichier, tandis que les nombres 120 et 26 janv. 15:56 représentent respectivement la taille en octets et la date de dernière modification du fichier.

Terminal : chmod 744 sauvegarde.sh

ls -l sauvegarde.sh

-rwxr--r-- 1 hajmohas ima-nogroup 120 26 janv. 15:56 sauvegarde.sh

./sauvegarde.sh

Sauvegarde effectuée dans

/home/h/hajmohas/INF203/sauve_TP1

il contient

total 32

drwxr-xr-x 25 hajmohas ima-nogroup 4096 30 janv. 2023 bottedefoin

-rwxr-xr-x 1 hajmohas ima-nogroup 4992 30 janv. 12:52 Candide_chapitre1.txt

-rw-r--r-- 1 hajmohas ima-nogroup 380 30 janv. 2023 copie2-fich1

-rw-r--r-- 1 hajmohas ima-nogroup 380 30 janv. 12:52 fich1

-rw-r--r-- 1 hajmohas ima-nogroup 380 30 janv. 12:52 fich3

drwxr-xr-x 8 hajmohas ima-nogroup 4096 30 janv. 12:52 programmes

drwxr-xr-x 2 hajmohas ima-nogroup 4096 30 janv. 12:52 scripts

[n] :

/INF203/TP1/scripts est maintenant notre répertoire de travail

[o] :

Les expressions $1, $2, $3 et $# représentent généralement les arguments dans un shell script (programmation de script en ligne de commande).

• $1 représente le premier argument passé au script

• $2 représente le deuxième argument passé au script

• $3 représente le troisième argument passé au script

• $# représente le nombre total d'arguments passés au script

En shell Linux/Unix, "$*" représente la liste des arguments passés à un script ou une commande shell. Chaque argument est considéré comme un élément distinct dans la liste. Cette expression peut être utilisée pour itérer sur chaque argument dans un boucle ou pour construire une commande à partir de la liste d'arguments.

[p] :

installeTP.sh $2

[q] :

```
#!/bin/bash

# usage: ./sauve_fichiers.sh rep suffixe motif

# set the parameters

rep_dir=$1

suffix=$2

pattern=$3

# create the save directory

save_dir="$rep_dir$suffix"

mkdir -p "$save_dir"

# copy all files (not directories) from the rep directory that match the pattern to the save directory

find "$rep_dir" -type f -name "*$pattern*" -exec cp {} "$save_dir" \;

# print a message indicating the save is complete

echo "Files matching pattern '$pattern' in '$rep_dir' saved to $save_dir"
```

[r] :

Terminal : grep Candide Candide_chapitre1.txt

c'est, je crois, pour cette raison qu'on le nommait Candide. Les

l'oracle de la maison, et le petit Candide ecoutait ses

Candide ecoutait attentivement, et

raison suffisante du jeune Candide, qui pouvait aussi etre

Elle rencontra Candide en revenant au

chateau, et rougit; Candide rougit aussi; elle lui dit

bonjour d'une voix entrecoupee, et Candide lui parla sans

comme on sortait de table, Cunegonde et Candide se

laissa tomber son mouchoir, Candide le ramassa, elle lui prit

cette cause et cet effet, chassa Candide du chateau a

grep "mm" Candide_chapitre1.txt

c'est, je crois, pour cette raison qu'on le nommait Candide. Les

honnete gentilhomme du voisinage, que cette demoiselle ne

croyait innocemment; car il trouvait Mlle Cunegonde

femme de chambre de sa mere, petite brune tres jolie

et tres docile. Comme Mlle Cunegonde avait beaucoup

comme on sortait de table, Cunegonde et Candide se

innocemment la main, le jeune homme baisa innocemment la main de

s'enflammerent, leurs genoux tremblerent, leurs

Il recherche la chaîne “mm” dans le fichier Candide_chapitre1.txt

Pour la commande grep "mmm" Candide_chapitre1.txt, il affiche rien parce qu’il n’existe pas « mmm » dans le fichier Candide_chapitre1.txt

Pour la commande grep "Ce" Candide_chapitre1.txt, il affiche rien parce qu’il n’existe pas «Ce » dans le fichier Candide_chapitre1.txt

Terminal : grep -v "a" Candide_chapitre1.txt

l'injure du temps.

" Il est demontre,

les experiences reiterees dont elle

leurs bouches se rencontrerent, leurs yeux

L’option -v de grep permet d’inverser la recherche, c’est-`a-dire de n’afficher que les lignes ne contenant pas la chaîne donnée

Donc il affiche seulement les ligne qui n’ont pas la chaîne « a ».

Terminal :grep -c Candide Candide_chapitre1.txt

10

grep -c Cunegonde Candide_chapitre1.txt

8

10 lignes contiennent le mot “Candide” dans ce fichier 8 lignes contiennent le mot “Cunegonde” .