

ITPI

10/05/2022 UNIX

Exercice sur ls:

→ Afficher le contenu de 1 bin

Pour afficher le contenu d'un rep on utilise la commande

ls nomrep

↳ ls 1bin

→ Afficher le contenu du rep courant sauf . et ..

↳ ls -A

do not list implied . and ..

NB

ls -a => affiche tous

les rep y compris . et ..

ls do not ignore rep courant parent

entries starting with .

→ L'option -R:

↳ Il nous permet de lister récursivement les rep courant et ses rep.

↳ son exécution:

ls -R

ls -a => il inverse le tri

→ Décrire le résultat ls -l.

↳ Cette commande affiche des info sur les fichiers & leur type.

ainsi résultat est sous forme de plusieurs champs

1 ^{er} champ	2 ^{er} champ	3 ^{er} champ	4 ^{er} champ	5 ^{er} champ	6 ^{er} champ	7 ^{er} champ
type, droit d'accès	Nombre lien physique	Nom du propriétaire	Nom du group	Taille en octets	mois	Jours
h de dernière modif	Nombre du fichier					

→ Tapissons Mkdir Mkdir.

↳ Cette commande permet de créer un rep qui s'appelle Mkdir.

→ Créons ~ / Rapport / docs / affaire et ~ / Rapport / docs / fait

Pour créer un rep et ses rep on utilise commande mkdir -p

Donc :
 [mkdir -p Rapport / docs / affaire] ~ => caractère spécial qui signifie le rep personnel
 [mkdir -p Rapport / docs / fait]
 (c'est pas la peine de l'utiliser (~) car on se trouve ds le rep perso)

→ Appel direct ds ~ / Rapport / docs / affaire :

cd ~ / Rapport / docs / affaire
chemin absolu → passer de cd .. / fait
chemin relatif
↓
fait et affaire
se trouvent ds m
m endroit
→ cd .. / fait
↳ se trouve ds docs

→ si on veut revenir à Rapport
↳ cd .. / .. /
→ Passer Rapport / docs
↳ cd . / docs.
→ Revenez rep prin
↳ cd ..

Exercices sur cp et mv:

cp

→ Que se passe-t-il si on fait cp -r essai tentative tentative3.

D'abord la commande sert à copier les fichiers et rep.

↳ Ds ce cas essai sera copié ds tentative et la copie gardera le m nom "essai"

Si tentative n'essai sera copié ds le rep courant et la copie sera nommée tentative.

→ faire une copie d'un fichier ds un rep ou à fichier de cendre

↳ Ds ce cas le fichier sera écrasé par notre copie

(on peut ajouter l'option -i pour le mode interactif pour que le syst nous demande une confirmation avant de l'écraser).

→ Possible de taper une ligne du genre de : cp f1 f2 f3 rep.

Oui, cette commande nous permet de copier f1 f2 et f3 ds rep ss changer leurs noms à cdtn que dernier arg soit rep.

Si on a un rep parmi les fichiers on doit \oplus l'option -r sinon que les fichiers seront copiés ms pas le rep

mv

→ mv f1 f2 avec f2 \exists .

si f2 \exists , il sera écrasé par f1 qui sera renommé

→ Renommons rep1 en rep2.

mv rep1 rep2

si rep2 \exists déjà rep1 sera déplacé à l'int de rep2 ss être renommé

→ Déplacer fichier et changer ss nom.

touch f1

mkdir rep1

mv f1 rep1/f2

\downarrow
 déplacer son nouveau nom

si f2 \exists déjà ds rep1 \rightarrow donc rep1 sera écrasé par notre nouveau fichier

Exercice sur Chmod :

→ Créons fichier txt file.

touch f1.txt

→ hc propriétaire et le groupe de file :

ls -l f1.txt

\hookrightarrow Pour moi le prop est mary123

et groupe : mary123

→ Ajoutons n au prop et au groupe.

↳ méthode symbolique | $rwx-rw-r--$ | ↳ méthode octale
chmod $u+n, g+n$ f1.txt ↓
pour moi chmod 774 f1.txt

→ Affectons maintenant droits du groupe aux autres :

↳ méthode symbolique :

chmod o=g f1.txt

→ Méthode octale

↳ Mkdir répertoire $rwx\ rwx\ rwx\ -\ -$
 $rwx\ r\ -\ -$

↳ chmod 740 répertoire

→ Il est aussi méthode symbolique :

chmod u=rwx, g=r, o=- répertoire

Exercices sur jokers :

→ En utilisant ls affichons la liste des fichiers du /bin dont les

noms :

* Commence par "a" et 2^{ème} lettre "b" ou "t"
n'importe quel caractère
ls /bin / a [b t] (*) → 2^{ème} lettre une des lettres
qui commence par a

* Contient "un" et se termine par "t"
contient

ls /bin / un * t
↓ → termine par "t"
n'importe quel caractère

* contient au moins 2 lettres "m"

ls /bin / * m * m *
↓ → 2 lettres m. n'importe la place de m

* Contient au moins 4 caractères

ls /bin / ???? (*) → au moins 4 ↑
↓ 4 caractères

* Conviient exactement 4 caractères

ls 1 bin ????? 4 caractères

* Commence et finit par un chiffre:

ls 1 bin | [0-9] * [0-9] | terminate
commence |

→ Supp les fichiers dont le nom se termine par chiffre:
nm * [0-9]

Exercice sur l'ordre : chercher des fichiers sur le disque

→ Cherchons tous les fichiers commençant par "Aouïa" suivi n'importe quel caractère et les terminé par chiffre entre 3 et 6.

func -name "[aA]" a "[3-6]"
nom de fichier

→ nom de fichier
→ Indiquons que fichier a été modifié plus de f_j , f_j , moins f_j
→ recherche la date dernière modification

Recherche
find |-mtime +7 → plus 7J
" " -7 → moins 7J
" " 7 → 7J

→ Specifications que le fichier recherché est un rep.

Find -type | d] → directory
↳ type dc Pichier

→ Cherchons les fichiers dont la taille est sup à une taille donnée

find -size (taille) [c] → on H → Hs
↓
recherche L → G1 → Gs
par taille L → W → Word

→ Utilisons find pour effacer tous les fichiers se terminant par .yc qui ont plus d'un mois.

find l -mtime +30 -name " ~" -exec rm {} \;

lsts les fichiers plus d'un mois nomme termine par ~

Exercices sur cat:

↳ affiche contenu de fichier

→ visualisons 2 fichiers d'un apnés l'autre avec cat :

cat β_1 β_2 affiche cont de β_1 et β_2

Exercice

→ Utilisons cat pour écrire un peu de txt ds $\beta_1 \rightarrow$ notes

→ on écrit des notes directement et on tape

ctrl d pour enregistrer

→ Obtenir un fichier β_2 qui correspond à β_1 dont les lignes

seraient désormais numérotées.

cat -n $\beta_1 \rightarrow \beta_2$

↳ pour numérotent les lignes

Exercices sur head et tail:

↳ début d'un fichier txt

→ Affichons ~~aff~~ 8

- 15 premières lignes du fichier etc / passwd
head -15 /etc /passwd

- 15 dernières ligne de m fichier:
tail -15 /etc /passwd

- 15 dernières lignes
tail -n -15 /etc /passwd

- ttes les lignes à partir de la 15:
tail -n +15 /etc /passwd

- des lignes 15 à 20

head -20 /etc /passwd | tail -n +15

→ Récupérons les lignes 5 à 9 d'un fichier de 12 lignes

head -9 | tail -n +5 →

5	6	7	8
9			

→ Affichons la 5^{ème} ligne d'un fichier:

head -5 | tail -1

↳ cette ligne travaille sur le résultat non pas sur le fichier.

Systèmes d'exploitation Unix

Travaux Pratiques n° 2

- 1) Chercher, à partir du répertoire courant, les fichiers réguliers dont les droits d'accès sont : rwxr-xr-x
find . -perm 755 -type f
- 2) Chercher, à partir du répertoire courant, les fichiers réguliers exécutables (c-à-d possédant x au niveau du propriétaire ou du groupe ou des autres).
find -perm /u=n, g=n, o=n
- 3) Chercher, à partir du répertoire courant, les fichiers réguliers possédant le droit w pour le propriétaire et la lecture pour le groupe et l'exécution pour les autres. u=w g=r, o=r
find -perm 241 -type f

Exercices sur grep

Les options de grep

1. Comment faire apparaître le numéro de la ligne où figure le mot recherché
2. Comment faire pour faire apparaître non pas les lignes où figurent le mot, mais les noms des fichiers ? grep -n nom fichier fichier
3. Comment faire apparaître les lignes où ne figure pas le mot recherché ? grep -v
4. Comment faire pour que grep ne recherche que les lignes où figure le mot tel quel, et non pas ses variantes ? Par exemple : on cherche le mot «travail», mais pas «travailleur» ou «travailler». grep -w recherche le mot exact

grep ' ^ bash .* bash \$ ' → commence et fini par bash.

Expressions régulières

1. Chercher toutes les lignes commençant par «a» ou «A». grep ' ^ [aA]' nom. grep -i ' ^ [aA]' nom
2. Chercher toutes les lignes contenant au moins un chiffre. grep ' [0-9]' nom
3. Chercher toutes les lignes commençant par une majuscule. grep ' ^ [A-Z]' nom
4. Chercher toutes les lignes finissant par un point d'exclamation. grep ' !\$' nom
5. Chercher toutes les lignes contenant au moins deux lettres «m» et peut être d'autres caractères grep ' m{2,}' nom
6. Chercher toutes les lignes contenant exactement deux lettres «m» et peut être d'autres caractères grep ' ^ [m]{2} \$' nom
7. Chercher toutes les lignes dont la deuxième lettre est «s» ou «t». grep ' .[st]' nom
8. Chercher toutes les lignes Contenant «un» et se termine par «t»; grep ' [un]{2}t\$' nom
9. Chercher toutes les lignes qui commencent et finissent par un chiffre. grep ' ^ [0-9].* [0-9]\$' nom
10. Comment éliminer les lignes vides dans un fichier ? grep ' .+' nom

Filtres et redirections

- sat -r /etc/ 1. Triez le fichier /etc/passwd par ordre alphabétique inverse et mettre le résultat dans file1
ls -l /etc/ 2. Lister le contenu de /etc triée par taille des fichiers et mettre le résultat dans file2
sat - 3. Combien de fichiers réguliers avez-vous dans votre répertoire courant?

/etc/passwd → contient des champs séparés par :
nom utilisateur : mdp : UID : GID : Info personnel :

grep ' ^ bash \$ ' → contient 1 exactement bash

4. Pour chaque ligne du fichier /etc/passwd, affichez :

- o Le cinquième caractère;
- o Les caractères 5 à 10, et le treizième;
- o Tous les caractères à partir du quinzième

grep ... /etc/passwd \Rightarrow 3 caract \Leftrightarrow ?? \rightarrow 3 caractères

grep ... \$ /etc/passwd \Rightarrow exactement 2 caractères

grep ... * /etc/passwd \Rightarrow plusieurs caractères (* \Rightarrow 0 ou plus caract)

grep ... + /etc/passwd \Rightarrow 1 ou plusieurs

grep -l "ensu" /etc/passwd \Rightarrow /etc/passwd

grep [a-zA] \rightarrow ne contient pas a ou A

grep [0-9]* \rightarrow 0 ou plusieurs chiffre

+ \rightarrow contient 1 ou plusieurs chiffre

grep ^[a-zA-Z] \rightarrow commence par chiffre ou alphabet

grep ^[0-9] [a-zA-Z]* \rightarrow commence par chiffre ou non.

Unix TP 2

→ Cherchons à partir du rep courant les fichiers réguliers dont les droits d'accès sont $rwnrnrn$.
droit d'accès

find -perm 755 -type f → Fichier régulier
↳ permission

→ Cherchons à partir du rep courant les fichiers rég exécutables:

find -perm /u=n, g=n, o=n -type f
↳ ou

find -perm /u+n, g+n, o+n -type f

find -perm /u=w, g=r, o=n -type f
les fichiers rég qui $u=w, g=r, o=n$

→ " "
find -perm -241
find -perm -u+w, g+r, o+n
find -perm -u=n, g=r, o=n

Expressions sur grep:

Les options de grep:

→ Apparition du numéro de la ligne où figure le mot recherché.
→ grep -A n → n lignes suivant le mot recherché.

→ " -B n → " avant "

→ " -C n → " avant et après "

→ Apparition non pas les lignes où figurent le mot mais les noms des fichiers.

→ grep -n nom fichier

→ Apparaîtront les lignes où ne figure pas le mot recherché.

→ grep -v nom fichier

→ Rechercher le mot tel quel et non pas de dérivées

→ grep -w mot nom fichier

Expressions régulières

→ Les lignes commençant par a ou A

grep '^ [aA]' nom fichier

→ Les lignes contenant au moins un chiffre:

grep '[0-9]' nom fichier

→ " par majuscule:

grep '^ [A-Z]' nom

→ " finissent par !

grep '! \$' nom

→ " contenant au moins 2 lettres "m":

grep 'm* m' nom

→ " exactement 2 "m"

grep '^ [^m]* m [^m]* m [^m]* \$'

→ " dont 2^{ème} lettre "sont"

grep '* [st]' nom

→ " Contenant un et se terminant par st.

grep 't[^n]* t \$'

→ " nom

- Les lignes qui commencent et finissent par un chiffre.
grep '^[0-9]' '\$' nom
- Éliminer les lignes vides.
grep '' nom.

Filtres et redirections:

- Trier le fichier /etc/passwd par nom et mettre résultat dans f1.
sort -r /etc/passwd > f1
↳ rediriger résultat de sort dans f1.

- Nombre de fichiers réguliers dans rep courant:

```
ls -l | grep -c '^-'
```

- affichons &
 - 5^{ème} caractère à cut -c5 /etc/passwd
découper fichier
 - car de 5 à 10 et le 13^{ème} à cut -c5-10,13 /etc/passwd
 - Tous cara à partir du 15^{ème} à cut -c15- /etc/passwd

Unix

TP3

- awk lit un txt ds un fichier, le modifie et envoie le résultat → la sortie standard
- le programme "awk" est une suite d'action de la forme : motif {action}
- permet de déterminer pour quels enreg est appliquée l'action.

Syntaxe : awk [-Fs] 'motif {action}' Fichier

- NR : Numéro de l'enreg courant ds fichier
- FS : séparateur de champ en entrée, " " par défaut
- NF : nbre de champ de l'enreg courant
- LP : Numéro de l'enreg courant ds l'entrée standard
- LP : sépa de champ pour sortie, " \n " par défaut
- RS : d'enreg pour la sortie, " \n " "
- RS : en entrée, " \n "
- RS : " "
- affichons nbre de lignes d'un fichier:
 - ligne courant ou dernière ligne
- awk 'END {print NR}' \$1
 - dernière ligne
- affichons 1er et dernier champ de chaque ligne:
 - awk ' {print \$1, \$NF}' \$1
- awk ' {print \$1, \$NF}' \$1
 - 1er et dernier champ de nom et la taille des fichiers du rep courant
- ls -l | awk '{ print "taille du fichier " \$NF " est " \$5}'
 - ls du rep courant
- ls -l | awk ',\$1{ print \$1, \$NF }'

→ " les lignes d'un fichier qui contiennent ab ou cde .

awk ' /ab/ || /cde/ } print \$' f1

→ " les lignes qui sont c entre 2 lignes = début et fin

awk '/^ensa/ ,/^Kali/ } print \$' /etc/passwd.

→ " taille totale des fichiers affichés par la commande ls .

ls -l | awk ' {s+=0} {s+=\\$5} END {print s}' f1 .

→ Remplacements champ 1 par 2 et 2 par 1 pour les UID >= 500

awk -F ":" '{OFS = ":"}{\$3 < 500 {print \$0}{\$3 >= 500

{var = \$1; \$1 = \$2; \$2 = var; print \$0}' f1

→ Remplacements de champs par "User nom de login" si il est φ

awk -F ":" '{OFS = ":"}{\$5 ~ /\$1/}{\$5 = "User nom de login"}

\$1; print \$0}' f1

→ Vérifions que 3^{ème} champ est entier: ☺

??

ls a* lancer des fich qui comm
par a et n'imp quel caract

ls *a* → contient au moins a

9m * .php → supp les fichi qui
se terminent par .php.

9m [0-9] → sup les fichiers
qui se terminent par chiffre
présente un seul caract (intervalle).

ls [a-z]*[0-9] → terminé par
comm par une lettre chiffre

ls [a-z][A-Z]*[0-9]
1^{er} 2^{ème}

ls [Aa-zA-Z]*[0-9]
maj ou min ds l'ordre

cd monusername → dossier personnel s'appelle monusername
ts les fichiers sont ordinaires.

on a 3 car on doit diviser pour les 3 catég lu/o/g

uncon mode octale : → chang pour les 3 cat (oblig)

chmod go-n f3 → tirer n pour f3. $\frac{10}{10}$

chmod go+wn f3 → aj wetu pour f3. $\frac{10}{10}$

chmod a-rw f3 → tirer rws pour les 3
all

chmod g=u f3 → chmod g=rw- f2
droit de u pour g → donner à g rrw

tree f1 → écrire ds f1 (vi)

les fichiers cachés sont des fichiers qui commence par .

• .rep courant /home/many123
parent du courant

• ls . la | grep -c ".d"

↳ compter les dossiers

• mkdir -p ~ /Rapport /docs /fait → TP unix

• " " -p ~ /Rapport /docs /fait → courant

• ls ~ /Rapport → contenu Rapport → ls -R ~ /Rapport

• tree ~ /Rapport → arborescence → sudo -s apt -get install tree

? : seul caractère * : plusieurs cara

gm ???.pdf → contient 3 caract exa

gm ???.*pdf → au moins 3 cara

[aA-zZ] → ideal

gm [eEz] * e ou z

gm Ee * e et z

[a-zA-Z]₁ [a-zA-Z]₂ *

gm -f * → sup ts les sys d'exp

gm /etc/usr/*[0-9]

pour. ↗ /etc /usr ↓ ↓
renv. ds ds
de racine

sudo -s : => root

ls a[st]* /bin

impossible

on a effacé

notes cours

19/05/2022

base name \Rightarrow name dirname : chemin
basename /home/mary123/file1 dirname _____ same
file1 /home/mary123

file a
ls /bin /a unt → terminate par un t

* -> rep 0 on plus ..

- * \rightarrow rep 0 ou plus
- * Is 1 bin / $\left[\frac{1}{2}, a\right]$ \rightarrow obli ne comme pas
negation. \leftarrow pour a

negation -
? un seul caractère n'importe lequel.

* contient un seul caractère m
 m ou $2m$.

• touch pq → si il n'existe pas il change la date de création.

• user est positionné dans la cd B de résultat ls -l
↳ ds l'examen (en général.)

• chmod a+w pq → w ds ts les catégories
" a-r pq → supp " "

• = → effacer chmod g=rw
↳ avoir rw ds groupe seulement

• find . → chercher ds fichier courant

• find /bin -type d ⇒ chercher les rep
ds bin

• find /etc -size +5 -type f

↳ chercher ds etc qui ont taille sup 5
Kilooctet → les autres majuscules.

find /var -type f -ctime +30

-name 'rc' -exec rm -f {} > /tmp/ → conf
passant

après commande unix

-rm (rc) → obligation

pour autoriser

mv -rc dest
mv -rc arg1 arg2

↳ supprimer les résultats
de find.