

Pràctica 8: Programes i processos

En aquesta pràctica veureu com podem fer que els processos que utilitzem es comuniquin els uns amb els altres, mitjançant la utilització de fluxos. D'altra banda, aplicareu els coneixements teòrics sobre filtres de la unitat. La pràctica és realitzarà tant en bash com en PowerShell (cal mostrar atenció a les indicacions de cada pregunta).

Comunicació entre processos i utilització de filtres

En aquesta part de la pràctica us remetem a les explicacions teòriques d'aquest nucli d'activitat per a que resolgueu les activitats que se us presenten..

1. Respecte el paginador **less**, indiqueu com es poden dur a terme les accions indicades a la taula de sota:

Accions	Less
Avançar pàgina a pàgina	barra espaciadora
Retrocedir una pàgina	ESC+V
Avançar línia a línia	Fletxa d'abaix
Retrocedir línia a línia	Fletxa d'adalt
Buscar una paraula, cap endavant	/paraula
Buscar una paraula, cap a darrera	?
Anar a una línia específica	gnum (g25)
Sortir	q

2. A **Linux**, escriuiu una comanda que mostri ÚNICAMENT els noms dels usuaris que tenen un nom que comença per 'm'.

Pista: Cal que trebal·leu amb el fitxer /etc/passwd.

```
sudo grep ^m /etc/passwd
```

3. A **Linux**, Quin efecte té la comanda **ls -l | tee primer.txt | sort -r > segon.txt** ? Quin diferència hi ha amb la comanda **ls -l | tee primer.txt | sort -R > segon.txt** ? Cal que expliqueu, pas a pas, què fa cadascuna de les comandes.

```
ls: llista de continguts d'un directori.
```

-l: permet obtenir més informació sobre els fitxers.
tee: mostra el progrés d'un procés mentre s'envien en les mateixes entrades als arxius de registre.
sort: ordena línies dels arxius d'entrada a partir dels criteris d'ordenació.
-r: inverteix l'ordre.
>: redirecciona la sortida estàndard d'un fitxer.
-R: llistat del contingut dels fitxers de forma recursiva.

4. Linux:

Escriviu una comanda que mostri el nom del segon fitxer més gran del directori /etc.

PowerShell:

Escriviu una comanda que mostri el nom del segon fitxer més gran del directori
C:\WINDOWS\system32\drivers\etc.

Linux: `du -hs* | sort -nr | head -2`

PowerShell: `Get-Childitem -r | select -first 2 C:\WINDOWS\system32\drivers\etc.`

5. A Linux, Escriviu una comanda que uneixi dos fitxers de text, un a continuació de l'altre, en un tercer.

`cat texte1.txt texte2.txt > texte3.txt`

6. Com podem copiar un fitxer sense utilitzar la comanda **cp**, utilitzant les redireccions de l'entrada i sortida estàndard? Copieu el fitxer C:\WINDOWS\system32\drivers\etc\hosts al vostre directori d'usuari.

`Copy-Item -Path C:\WINDOWS\system32\drivers\etc\hosts -Destination C:\Users\apere`

7. A Linux, com podríem construir una **pipe** o canonada (|) amb els redirectors de l'entrada i sortida estàndard (< >)? Cal utilitzar un fitxer temporal que hem d'esborrar després del seu ús. Escriviu l'equivalent a **ls -l | wc -l**.

```
user@host:~$ cat newfile.txt
cat: newfile.txt: No such file or directory
user@host:~$ echo "Here is some text" > newfile.txt
user@host:~$ cat newfile.txt
Here is some text
user@host:~$ echo "and some more text" >> newfile.txt
user@host:~$ cat newfile.txt
Here is some text
and some more text
user@host:~$ echo 'Oops just new text now' > newfile.txt
user@host:~$ cat newfile.txt
Oops just new text now
```

8. A **Linux**, Escriuiu una comanda basada en cat que tingui el mateix comportament que **nl**.

cat -n text.txt

9. **Linux**:

Escriuiu una comanda que mostri un llistat de tots els usuaris del sistema (ÚNICAMENT el nom), ordenats per ordre alfabètic.

Pista: Cal que treballem amb el fitxer **/etc/passwd**.

PowerShell:

Escriuiu una comanda que mostri un llistat de tots els processos del sistema (ÚNICAMENT el nom), ordenats per no de PID.

Linux: cut -d -f1 /etc/passwd |sort

Powershell: Get-Process | Select-Object Name, Id, Company | Sort-Object Name

10. **Linux**:

Intenteu crear un directori a l'arrel del vostre sistema (/) i envieu la **sortida d'error** d'aquesta comanda a un fitxer del vostre directori d'usuari. Què passa? Proveu el mateix amb la **sortida estàndard**, què passa? Com podríem fer que aquesta **sortida d'error NO es mostrés per pantalla ni es guardes a cap fitxer**?

PowerShell:

Intenteu llistar un fitxer que no existeixi i emmagatzemeu la sortida d'error a un fitxer 'error.txt'. Com podríem fer que aquesta **sortida d'error NO es mostrés per pantalla ni es guardes a cap fitxer**?

11. **Linux**:

Executeu una comanda que us mostri **ÚNICAMENT les línies repetides**, de forma consecutiva, del fitxer **give_it_away.txt**.

Executeu una comanda que us mostri el nombre de repeticions de cadascuna de les línies del fitxer give_it_away.txt.

PowerShell:

Executeu una comanda que mostre tots els tipus de comandes (cal fixar-se amb la columna '**Source**') que trobem a PowerShell (utilitzeu la comanda **Get-Command**)

Per visualitzar únicament les línies repetides executarem la comanda `$ sort give_it_away-txt | uniq -d`
Per visualitzar el nombre de repeticions de cada línia utilitzarem la comanda `$ sort give_it_away-txt | uniq -c`

12. A **Linux**, Utilitzeu la comanda `paste` per a unir tres fitxers. Feu que tots els tabuladors retornats es converteixin en espais. Feu tota aquesta interacció en una única execució.

Per unir els tres documents utilitzarem la comanda `-m` i per poder reemplaçar els tabuladors per espais utilitzarem la comanda `-t','`
`sort -m texte1.txt texte2.txt texte3.txt|-t','`

13. A **Linux**, Quina opció de la comanda `tr` permet comprimir **caràcters repetits consecutivament** en un de sol. Proveu aquesta comanda amb les “o” de la següent cadena “boomboompooowoooot”. Com ho faríem si en compte de voler comprimir caràcters repetits ho féssim per a comprimir **espais consecutius** (comprimiu doncs els espais d'una sortida de la comanda `'ls -l'`)?

`cat boomboompooowoooot.txt | tr -s “`

14. A **Linux**, Escriviu una comanda que, **per a cadascun dels directoris de l'arrel del sistema**, crei un fitxer de text amb el seu nom a l'interior (en format text). Els fitxers de text generats hauran de començar amb el prefix `'carpeta_'`.

15. A **Linux**, Escriviu una línia, per a cada apartat, que us permetin:

- a) Crear un directori i, si s'ha pogut crear correctament, mostrar el missatge “directori creat correctament”.
- b) Crear un directori i, si no s'ha pogut crear correctament, mostrar el missatge “no s'ha pogut crear el directori”.

Per a comprovar el funcionament d'aquests apartats intenteu crear les carpetes al vostre directori d'usuari i/o a l'arrel del sistema.

```
#!/bin/bash
for dir in $@
do
    if [ -d $dir ]
    then
        echo “La carpeta $dir ya existe.”
    Else
        mkdir $dir
        if [ $? -eq 0 +
        then
            echo “$dir se ha creaco con éxito”
        else
```

```
echo "Ups! Algo ha fallado al crear $dir"  
fi  
fi  
done
```

16. A **Linux**, Quin efecte té la comanda **echo "-----" | cat fitxer1 - fitxer2** ? Quin efecte té la comanda **cat fitxer1 - fitxer2 < fitxer3** ?

Copia les dades del fitxer1 i fitxer2 en el nou fitxer3

17. Linux:

Els fitxers de configuració solen indicar els comentaris amb línies que comencen amb **#**.
Escriuiu una comanda que mostri totes les línies de comentaris del fitxer **grub.cfg** (indiqueu la ruta completa).

Escriuiu una comanda que mostri el contingut del fitxer **grub.cfg** eliminant els comentaris i també les línies buides.

Pista: una línia buida és aquella que no conté cap caràcter.

PowerShell:

Escriuiu una comanda que mostri el contingut del fitxer **C:\WINDOWS\system32\drivers\etc\networks** eliminant els comentaris i també les línies buides.

Linux: `grep -v "#" grup.cfg | sed -e '/^$/d'`
PowerShell:

18. Preneu l'arxiu de text **give_it_away.txt** i realitzeu les accions següents (en el mateix ordre, comunicades amb canonades o pipes):

Linux:

1. Elimineu les línies repetides consecutives.
2. Canvieu totes les lletres majúscules que hi hagi per minúscules.
3. Elimineu totes les línies buides que hi hagi.
4. Ordeneu les línies resultants en ordre alfabètic invers.
5. Numereu totes les línies.
6. Quedeu-vos només amb les línies que van de la 10 a la 40.
7. Elimineu la numeració de les línies que havíeu afegit al pas 5.
8. Reordeneu les línies en ordre alfabètic per la segona paraula de cada línia.

PowerShell:

1. Elimineu les línies repetides consecutives.

2. Canvieu totes les lletres majúscules que hi hagi per minúscules.
3. Elimineu totes les línies buides que hi hagi.
4. Ordeneu les línies resultants en ordre alfabètic invers.
5. Numereu totes les línies.
6. Quedeu-vos només amb les línies que van de la 10 a la 40.

Expliqueu cadascun dels passos que hagueu utilitzat en aquesta comanda.

Linux:

1. `uniq give_it_away.txt`
2. `tr '[:lower:]' '[:upper:]' give_it_away.txt`
3. `sed -i '/^$/d' give_it_away.txt`
4. `sort -r give_it_away.txt`
5. `nl -b a give_it_away.txt`
6. `sed -n '10,40p;41q'`
7. `awk '{ $1="" }; print $0'`
8. `sort -k2 give_it_away.txt`

PowerShell:

19. Tant a **Linux** com a **Windows**, escriviu una comanda que retorni la MAC de la nostra tarjeta de xarxa. Per a fer-ho, caldrà que:
 1. Executeu la comanda **ifconfig** ('**ipconfig** /all' o '**Get-NetAdapter**' a PowerShell)
 2. Seleccioneu d'alguna manera la línia que conté la MAC de la vostra tarjeta de xarxa.

Nota: A **Linux** seleccioneu únicament el valor de la MAC, a **Windows** podem obtenir tota la línia sencera

Expliqueu cadascun dels passos que hagueu utilitzat en aquesta comanda.

```
ifconfig | grep ether
```

20. Tant a **Linux** com a **Windows**, escriviu una comanda que mostri ÚNICAMENT els PID dels 5 processos que consumeixen més memòria del sistema.

Nota: en el cas de Windows ens referim a la **memòria paginada**

```
ps -aux | awk '{print $2, $4, $11}' | sort -k2r | head -n
```

21. Escriviu una comanda que permeti mostrar per pantalla el missatge següent:
*****MICRO\$OFT RULE\$***.**

```
write -n console  
Enter  
Escrivim el missatge  
Ctrl+D
```

22. A **Linux**, Per a què serveix cadascuna de les expressions regulars de **grep** que s'indiquen a continuació?

Comanda	Resultat
^	Cercar per l'inici de línia.
\$	Cercar pel fi de línia.
[...]	Qualsevol caràcter central de la llista.
[^...]	Que comenci per aquest valor.
[m-r]	Entre m i r.
.	Qualsevol caràcter únic.
c*	
.*	Zero o més aplicacions de caràcter únic.
\{n\}	Indica que el caràcter o expressió s'ha de rebre exactament x vegades.
\{n, \}	Indica que el caràcter o expressió s'ha de repetir x vegades o més.
\{n,m\}	Indica que el caràcter o expressió anterior pot existir entre x y z vegades.
\	Ignora el significat especial del pròxim caràcter

23. A **Linux**, Expliqueu què fa cadascuna de les comandes següents:

Comanda	Resultat
grep give *	Mostra totes les línies que contenen la cadena give en una llista d'arxius.
grep -l give *	Mostra el nom dels dels arxius.
grep -i give *.txt	Mostra el nombre de frases en les que surt give dins qualsevol fitxer de text.
grep -i -h give *.txt	Mostra el mateix que en l'anterior comanda però sense el nom del fitxer.
grep -n give give_it_away.txt	Mostra el número de la línia amb les paraules sol·licitades.

<code>grep -cv give give_it_away.txt</code>	Respon amb un número, que no se el que representa.
<code>grep give\$ give_it_away.txt</code>	Sense resposta.
<code>grep ^give\$ *</code>	Sense resposta.
<code>grep [A-Z] give_it_away.txt</code>	Marca totes les majúscules escrites al fitxer.
<code>grep 'Stal\{1,2\}man' *</code>	Sense resposta.
<code>grep '^error' *.log > problemes.txt grep '^dump' *.log >> problemes.txt</code>	Sense resposta.
<code>grep -c /bin/bash /etc/passwd</code>	Conta la quantitat de vegades que apareix una entrada repetida en ambdos fitxers.