

# Exercicis Scripts Usuaris

Realitza les següents funcions (ja fetes a classe)	1
Fer aquestes funcions noves:	1
Exercicis de funcions avançats:	2
Exercicis de processar fitxers línia a línia / stdin	2
Exercici training	3
Exercicis shadow	3

Realitza les següents funcions (ja fetes a classe)

**1. *showUser login***

Rebut un login per argument mostrar camp a camp les dades de l'usuari (etiqueta i valor del camp). Validar que es rep almenys un argument. Validar que és un login vàlid.

**2. *showUserGecos login***

Rebut un login per argument mostrar camp a camp les dades del GECOS de l'usuari (etiqueta i valor del camp). Validar que es rep almenys un argument. Validar que és un login vàlid.

**3. *showGroup gname***

Rebut un gname per argument mostrar camp a camp les dades del grup (etiqueta i valor del camp). Validar que es rep almenys un argument. Validar que és un gname vàlid.

Fer aquestes funcions noves:

**4. *showUser login***

Modificar la funció showUser feta anteriorment per fer que mostri també el gname de l'usuari.

**5. *showUserList login[...]***

Realitza el mateix que la funció showUser però processa n lògins que rep com a argument. Validar que el login és vàlid i que n'hi ha almenys un.  
Atenció, feu tot el codi de nou, sense cridar per res la funció showUser.

**6. *showUserIn < fileIn***

Mostra la informació del usuari tal i com feia showUser o showUserList però processa l'entrada estàndard. Cada línia de l'entrada és un login.

### **7. *showGroupMainMembers gname***

Donat un gname mostrar el llistat dels usuaris que tenen aquest grup com a grup principal. Com a llistat mostrar el login, uid, dir i shell dels usuaris (en un format de llistat del grep). Validar que es rep un argument i que el gname és vàlid.

### **8. *showGroupMainMembers gname***

Ampliar l'exercici anterior fent modificacions en el format de presentació de les dades:

- a) Separades per un tabulador (cada camp d'un usuari)
- b) Separat cada camp per dos espais.
- c) Ordenat per login i separat per dos espais.
- d) Ordenat per uid, separat per un espai i tot en majúscules.

## Exercicis de funcions avançats:

### **9. *showAllGroupMainMembers***

Llistar per ordre de gname tots els grups del sistema, per a cada grup hi ha una capçalera amb el nom del grup i a continuació el llistat de tots els usuaris que tenen aquell grup com a grup principal, ordenat per login.

### **10. *showAllShells***

Llistar totes els shells utilitzats per usuaris del sistema per ordre alfabètic i per a cada shell llistar els usuaris que l'utilitzen. De cada usuari mostrar el login, uid, gid i home. Els usuaris es llisten per ordre de uid.

### **11. *showAllGroupMainMembers2***

Fer una nova versió de *showAllGroupMembers* on per cada grup es llisti una capçalera amb el nom del grup i la quantitat d'usuaris que tenen aquest grup com a grup principal. Per a cada grup llavors es llisten les línies de detall dels usuaris que hi pertanyem, per ordre de login, mostrant login, uid, home i shell.

### **12. *showAllShells2***

Fer una nova versió de *showAllShells* que per a cada shell mostra una capçalera amb el nom del shell i la quantitat d'usuaris que l'utilitzen, i tot seguit el llistat d'aquests usuaris.

Atenció, cal filtrar els shells. No es mostren tots sinó només aquells que tinguin dos o més usuaris assignats.

## Exercicis de processar fitxers línia a línia / stdin

### **13. *numStdin***

Numerar línies de stdin. Aquest és un exercici de repàs de conceptes antics: iterar línia a línia tsdin.

### **14. *numFile file***

Numerar línies d'un file rebut com a argument. Validar que s'ha rebut un argument i també que l'argument rebut és un regular file que existeix.

### **15. *numStdin [file]***

Numerar línies d'un file rebut com a argument o de stdin si no s'ha indicat cap argument. Si s'ha rebut un argument validar que sigui un regular file que existeixi. Cal validar el número d'arguments rebuts (cap o un).

Recomanació, establiu sempre incondicionalment com a valor per defecte a processar stdin i modifiqueu-lo al fitxer rebut únicament si s'ha passat un argument.

#### **16.filterGid file**

Processa un fitxer rebut com a argument que té un format de línies similar al /etc/passwd. Cal mostrar per stdout només les línies dels usuaris amb gid >= 500. Cal validar que s'ha rebut un argument i que és un regular file existent.

#### **17.filterGid [file]**

Processar línies de tipus /etc/passwd i mostrar només les línies de gid >= 500. Les línies a processar poden ser les de un fitxer rebut com a argument o les de stdin. Validar el número d'arguments rebuts (cap o un) i si s'ha rebut un argument que sigui un regular file

#### **18.showGidMembers [file]**

Processa un fitxer de text que conté un gid per línia o bé stdin (on també espera rebre un gid per línia). Mostra el llistat dels usuaris (login, uid, home) d'aquells usuaris que tenen aquest grup com a grup principal.

#### **19.showGidMembers2 [file]**

Processa un fitxer de text que conté un gid per línia o bé stdin (on també espera rebre un gid per línia). Mostra el llistat dels usuaris (login, uid, home) d'aquells usuaris que tenen aquest grup com a grup principal. Fa un having, mostrant només els grups si hi ha almenys 3 usuaris al grup.

### Exercici training

#### **20.showPedidos oficina**

Donat un número d'oficina com a argument (validar es rep un argument i que és un número d'oficina vàlid), obtenir els codis dels representants de vendes que treballen en aquesta oficina i llistar les comandes d'aquest repventas. És a dir, estem fent un llistat de les comandes agrupades per venedors dels venedors que treballen en una determinada oficina.

### Exercicis de funcions amb crides a funcions

#### **21.getHome login**

Fer una funció que rep un login (no cal validar que rep un argument ni que sigui un login vàlid) i mostra per stdout el home de l'usuari.  
La funció retorna 0 si ha pogut trobar el home i diferent de zero si no l'ha pogut trobar.

#### **22.getHoleList login[...]**

Fer una funció que rep un o més login d'arguments i per a cada un d'ells mostra el seu home utilitzant la funció getHome. Cal validar els arguments rebuts.

#### **23.getSize homedir**

Donat un homedir com a argument mostra per stdout l'ocupació en bytes del directori. Per calcular l'ocupació utilitza du del que només volem el resultat numèric. Cal validar que el directori físic existeix (recordeu que hi ha usuaris que poden tenir de home valors que no són rutes vàlides com /bin/false). Retorna 0 si el directori existeix i un valor diferent si no existeix. Cal usar les funcions que estem creant.

#### **24.getSizeIn**

Ídem exercici anterior però processa un a un els login que rep per stdin, cada línia un lògin. Valida que el login existeixi, si no és un error. Donat el login mostra per stdout el size del seu home. Cal usar les funcions que estem definint.

#### **25.getAllUsersSize**

Processa una a una les línies del fitxer /etc/passwd i per a cada usuari mostra per stdout el size del seu home.

### Exercicis shadow

#### **26.showShadow login**

Donat un login mostrar els camps del /etc/shadow d'aquest login, mostrant camp a camp amb una etiqueta i el valor del camp. Validar que es rep un argument i que el login existeix.

#### **27.showShadowList login[...]**

Mostrar per a cada lògin les dades del seu shadow. Validar que es rep almenys un argument i que existeix cada un dels lògins.

#### **28.showShadowIn < login**

Mostrar els camps del shadow de cada usuari rebut per l'entrada estàndard. Cada línia de l'entrada estàndard correspon a un login. Verificar que cada un dels lògins existeix.