FONAMENTS D’ENGINYERÍA

Pràctica 1

Arnau Marcos Almansa

Jairo Abdul Villarroel

Grup 45

**Objectius de la pràctica**

L’objectiu d’aquesta pràctica és aprendre i familiraitzar-se amb la programació en *bash*. En concret, familiaritzar-se amb la sintaxi del llenguatge, algunes comandes essencials i el tractament de fitxers i dades.

**Plantejament**

La practica consisteix en el desenvolupament d’un script en *bash* que permeti que un usuari pugui obtenir dades d’un fitxer *csv*. Aquest fitxer contindrà un llistat de series i pelicules. L’usuari ha de poder, mitjançant diversos menús, obtenir un registre aleatori del fitxer, aplicar un filtre de puntuació (rating) i interactuar amb un sistema de preferències que seràn utilitzades a l’hora de seleccionar el registre aleatori.

Per desenvolupar aquest script hem seguit les pautes de l’enunciat. Cada opció del script s’ha programat independentment en funcions separades que es criden desde el bucle principal o desde altres funcions. D’aquesta manera, ens queda un codi modular i reutilitzable.

**Descripció funcional del script**

En iniciar el *script*, s’agafa tot el contingut del fitxer *netflix.csv* excepte la primera línia, s’eliminen els duplicats i s’introdueix a un nou fitxer *netflix\_unique.csv*.

Si es detecta el fitxer de preferències, es generarà un fitxer *filtrat\_preferencies.csv* que contindrà les series i pel·licules que compleixin les preferències de l’usuari.

La base del *script* és un bucle amb una estructura de control *case*. Això fa la funció de menú que, segons l’opció que seleccioni l’usuari, crida a una funcio o a una altra.

Si es selecciona la opció 5, el programa finalitza.

Les funcions principals son: *recomanacio\_rapida*, *llistar\_per\_any*, *llistar\_per\_rating* i *criteris\_de\_cerca*.

A *recomanacio\_rapida*, mostrem per pantalla una sèrie/pel·lícula aleatòria. Si hi ha preferències, *recomanacio\_rapida* només utilitzarà les series/pel·lícules que coincideixin amb els criteris de cerca.

Si el fitxer amb les series filtrades per preferències no té registres, avisarà a l’usuari.

Per escollir un numero de registre aleatori, generem un número aleatori i calculem el residu d’aquest entre el numero de linies del fitxer i li afegim 1. D’aquesta manera ens queda un número entre l’1 i la longitud del fitxer que podem utilitzar per obtenir el registre.

La funció *llistar\_per\_any* llegeix un any introduït per l’usuari i l’utilitza per mostrar per pantalla un llistat amb totes les series/pel·lícules que hagin sigut publicades aquell any.

Per fer el filtre per any hem utilitzat la comanda *grep*, aquesta comanda ens permet filtrar les linies que compleixin una expressió regular. En aquest cas, busquem els registres que tinguin l’any escollit a l’antepenúltim camp. Això és així per tractar correctament certs registres que inclouen una coma (el separador de camps) al títol o a la descripció. Si l’espressió regular estigues escrita per filtrar per la cinquena columna desde el principi de la línia en comptes de l’antepenúltima, els registres anteriors es filtrarien incorrectament.

*La funció llistar\_per\_rating* mostra un menú on l’usuari pot escollir un número d’estrelles per filtrar o sortir. La funció mostrará per pantalla les series/pel·lícules que tinguin el mateix nombre d’estrelles escollides per l’usuari o més.

Per fer això, quan l’usuari introdueix un numero, s’assigna un rating mínim i un rating máxim i es filtren les series que estiguin entre aquests dos ratings. Un cop s’han filtrat les series, s’ordenen per rating de menor a major i els donen format per mostrar-les. En donar format, es reemplaça el camp del rating numeric del 0 al 100 per un rating d’estrelles del 1 al 5. Finalment es mostren per pantalla.

La ultima funció, *criteris\_de\_cerca*, obre un menú on l’usuari pot escollir entre modificar les preferències, eliminar-les, o mostrar-les.

Si escull modificar les preferències, haurà de seguir les instruccions i introduir els valors per a cada criteri (any, rating, stars), si no introdueix cap valor a un criteri, aquest serà ignorat.

Si l’usuari escull l’opció d’eliminar les preferències, s’eliminarà el fitxer on s’emmagatzemen els valors de les preferències.

Si escull la tercera opció, mostrarà les preferències actuals, amb el seu nom i el seu valor.

**Problemes**

Un dels primers problemes que vam tenir, que era part de l’enunciat, va ser que el document *netflix.csv* tenia moltes series/pel·lícules repetides. Això ho hem arreglat amb la següent comanda:

tail +2 netflix.csv | sort -u > netflix\_unique.csv

Això treu la capçalera del fitxer amb el *tail +*2 i elimina registres repetits amb *sort -u*. Al treure la capçalera, evitem que es barregi amb la resta de registres en eliminar repetits i ens facilita tractar el fitxer a diferents parts del programa, com quan mostrem un registre aleatori, on no hem de preocuparnos de que ens mostri la capçalera per accident.

Un altre problema que vam tenir, va ser quan feiem el filtre a *llistar\_per\_any.* Vam observar que la següent comanda:

grep “$any” netflix\_unique.csv | cut -d',' -f1,2 | less

mostrava registres que tenien el valor $any a camps que no eren el d’any, sino el titol, la descripció o el rating.

Això ho hem solucionant canviant la comanda anterior per:

grep ",$any,[0-9]\*,[0-9]\*$" netflix\_unique.csv | cut -d',' -f1,2 | column -t -s "," | less

Fent això ens assegurem que la informació l’agafarà de l’antepenúltima columna del document netflix\_unique.csv, que es l’any de publicació. Si no féssim això, la cerca es faria per qualsevol camp.

Un problema bastant interessant que ens ha succeit ha sigut que, quan utilitzavem el valor d’una variable que contenia espais o caracters separadors sense posar-la entre cometes dobles, aquesta variable es convertia en tantes variables com paraules separades entre espais o altres caracters.

Per exemple, si la variable *frase* conté «Hello World» i passem *$frase* (sense comente) com argument a una funció, «Hello» será el primer argument i «World» serà el segon. No s’enviaràn com un únic argument.

Això ens ha passat a diversos punts del script i ens ha obligat a buscar-ho a internet fins a entendre que, quan una variable conté multiples paraules separades per espais o altres caracters, cal expandir aquesta variable entre cometes dobles o es poden produir errors.

**Execució**

L’*script* es interpretat, i per executar-lo l’hem de cridar en un terminal *bash*. Abans d’executar-lo cal donar-li permisos d’execució si no els te amb la comanda *chmod*.

**Conclusions**

La practica ha sigut molt didàctica. Ha estat el suficientment guiada per a no perdre el fil i el suficientment lliure per a haguem necessitat buscar informació a internet.

La part de programació ha sigut curta ja que una part important del temps l’hem invertit investigant la sintaxi de *bash*, comandes i solucionant problemes que ens han sorgit per falta de coneixement sobre aquest llenguatge de scripting.

Després d’haver terminat aquesta pràctica, creiem que no hem adquirit un coneixement massa exaustiu sobre bash, però si el suficient per a poder dur a terme projectes més complexos en un futur. També hem aprés com buscar informació a internet en cas de tenir un dubte o problema i creiem que això és una part essencial per a poder continuar aprenent de forma independent i per a projectes futurs inclús d’altres assignatures.