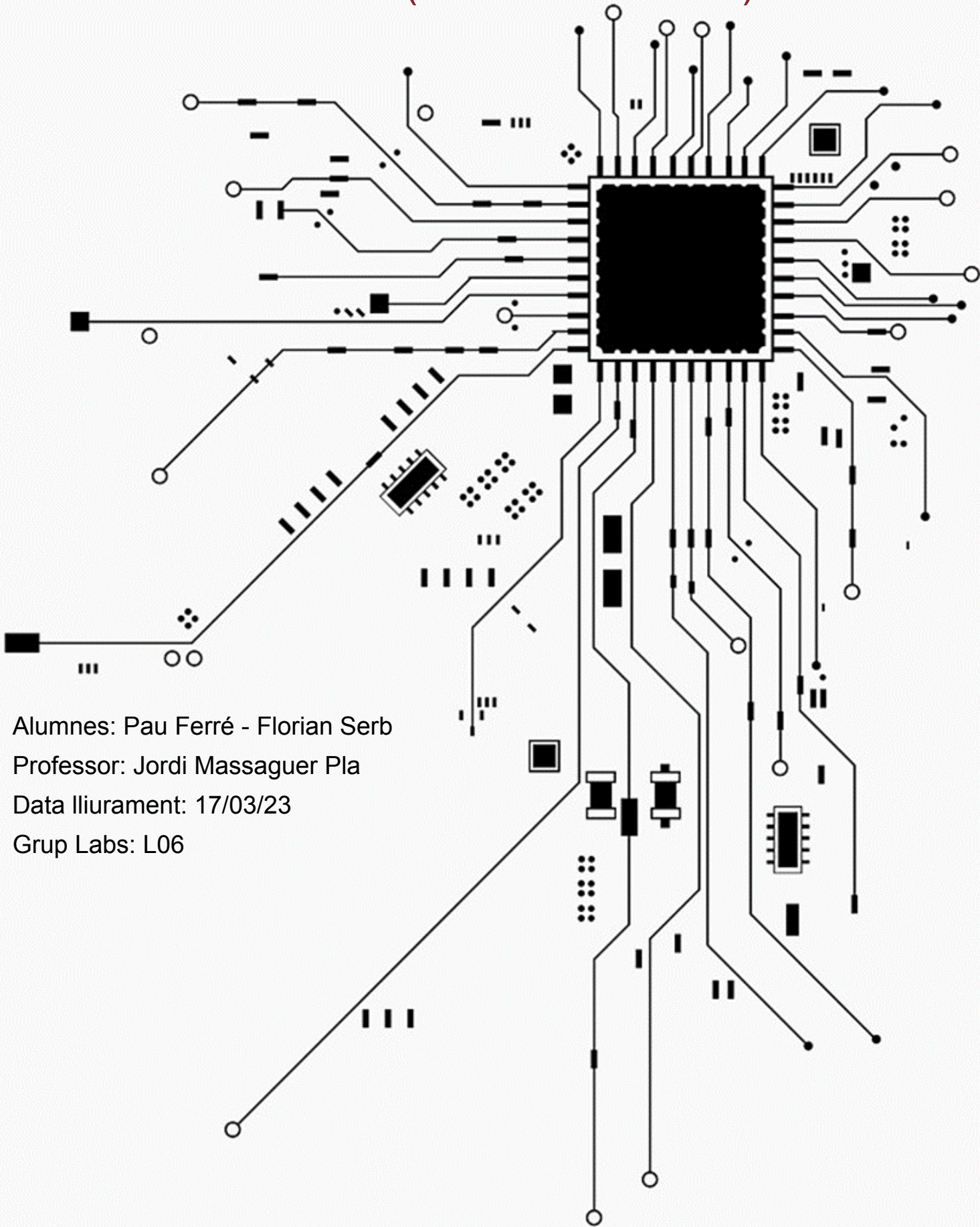


PRÀCTICA 1 (BASH&PYTHON)



Alumnes: Pau Ferré - Florian Serb

Professor: Jordi Massaguer Pla

Data lliurament: 17/03/23

Grup Labs: L06

Índex

Introducció	4
1. Bash tipus.sh	5
1.1 Disseny	5
1.2 Codi	5
1.3 Joc de proves	6
2. Bash compfitxer.sh	7
2.1 Disseny	7
2.2 Codi	7
2.3 Joc de proves	9
3. Bash mesactual.sh & Python mesactual.py	10
3.1 Mesactual.sh	10
3.1.1 Disseny	10
3.1.2 Codi	10
3.1.3 Joc de proves	11
3.2 Mesactual.py	11
3.2.1 Disseny	11
3.2.2 Codi	11
3.2.3 Joc de proves	12
4. Bash compdir.sh	13
4.1 Disseny	13
4.2 Codi	13
4.3 Joc de proves	16
5. Bash comptot.sh & Python comptot.py	17
5.1 Comptot.sh	17
5.1.1 Disseny	17
5.1.2 Codi	17
5.1.3 Joc de proves	23
5.2 Comptot.py	24
5.2.1 Disseny	24
5.2.2 Codi	24
5.2.3 Joc de proves	29
6. Bash creanou.sh	29
6.1 Disseny	29
6.2 Codi	30
6.3 Joc de proves	30
7. Bash p1.sh & Python p1.py	32
7.1 P1.sh	32

7.1.1 Disseny	32
7.1.2 Codi	32
7.1.3 Joc de proves	33
7.2 P1.py	33
7.2.1 Disseny	33
7.2.2 Codi	34
7.3.3 Joc de proves	35
Conclusió	36

Introducció

En aquest treball hem realitzat una sèrie d'scripts i d'arxius python. Aquests arxius s'utilitzaran per a poder realitzar una sèrie de funcions les quals estan recopilades als arxius p1.sh i p1.py. Els requisits per a que les funcions funcionen correctament estaran explicades en l'apartat de disseny de cada apartat.

Tenim diversos arxius:

- **tipus.sh**: Tipus de fitxer passat per l'usuari
- **compfitxer.sh**: Comparació entre dos arxius. Retorna un percentatge de similitud i per canal d'error passarà les línies incorrectes.
- **mesactual.sh** i **mesactual.py**: Comparació entre dos arxius. Retorna la ruta absoluta de l'arxiu modificat més actual.
- **compdir.sh**: Comparació entre dos directoris, tenin en compte els arxius dins de cada directori. Retorna un percentatge de similitud, les línies diferent per canal d'error i passa la ruta absoluta dels arxius modificats més actuals dins de recents.log
- **comptot.sh** i **comptot.py**: Comparació entre 2 directoris tenin en compte els subdirectoris i els subarxius de l'interior. Retorna un percentatge de similitud, les línies diferents per canal d'error i passa la ruta absoluta de tots els subarxius dels subdirectoris més actuals dins de recents.log
- **creanou.sh**: Copia tots els arxius guardats a recents.log dins d'una carpeta que passa l'usuari.
- **p1.sh** i **p1.py**: Script el qual comparar dos directoris o dos arxius i els guardarà a una ultima carpeta. Retorna el percentatge de similitud, pel canal d'error es pasaran les línies diferents, dins de recents.log es guardarà els arxius modificats més actuals i finalment aquest arxiu es copiaran a un directori passat per l'usuari.

1. Bash tipus.sh

Crearem una funció bash tipus.sh que donat un arxiu que li passem per paràmetre ens retorna 1 si és de tipus fitxer, 2 si és de tipus directori i 3 si és de qualsevol altre tipus. (retorn de l'script, no per la sortida estàndard).

1.1 Disseny

En aquest primer apartat hem decidit, utilitzar **\$#**, ja que d'aquesta manera agafem tot el que l'usuari introdueix per teclat i si aquest no introdueix un directori o fitxer o alguna cosa li mostrarem el següent missatge d'error, i a més a més redirigim l'error amb **>&2**.

A continuació identifiquem que és el que ha introduït l'usuari, si és un fitxer ho identifiquem amb el **-f** i retornem el número 1, si és un directori ho identifiquem amb un **-d** i retornem el número 2, i per últim si és qualsevol altra cosa, per exemple un fitxer sense extensió o un .mp4 o qualsevol altra extensió que no sigui .txt, retornarem el número 3.

En el cas en que l'usuari afegeix un fitxer que no existeix, llavors el script retornarà 3 ja que és qualsevol altre a arxiu i directori.

1.2 Codi

```
#!/bin/bash
```

```
#Condicional per a que no pugui afegir un número diferent a 1 fitxer.
```

```
if [ $# -ne 1 ]; then
```

```
    echo "Has de afegir 1 fitxer">&2
```

```
    exit
```

```
fi
```

```
fit=$1
```

```
if [ -f $fit ]; then
```

```
    exit 1
```

```
elif [ -d $fit ]; then
```

```
    exit 2
```

```
else
```

```
    exit 3
```

```
fi
```

1.3 Joc de proves

Descripció	Característiques	Resultat
L'usuari introdueix un fitxer.	Usuari: fitxer.txt	Programa: 1 el programa retorna el número 1 indicant que és un fitxer.
L'usuari introdueix un directori.	Usuari: practica	Programa: 2 el programa retorna el número 2 indicant que és un directori.
L'usuari introdueix un fitxer sense cap mena d'extensió.	Usuari: c	Programa: 3 el programa retorna el número 3 indicant que és qualsevol altra cosa, que no és ni fitxer ni directori.
L'usuari introdueix un fitxer .mp4.	Uusari: vid.mp4	Programa: 3 el programa continua retornant el número 3, ja que no és cap directori i tampoc és un fitxer .txt

2. Bash compfitxer.sh

2.1 Disseny

En aquest script, hem de tenir en compte que alhora d'executar l'script tot sol, per a no eliminar la informació que ja està guardada al document error.txt hem d'afegir al final 2>>error.txt.

Si en els dos documents tenim una línia exactament igual però en diferent posició (hola, fitxer1: línia 3, fitxer2: línia 6), llavors dins d'error.txt el format de les línies quedarà un espai blanc. Alhora de realitzar els càlculs no hi ha cap problema.

També hem de tenir en compte que si no afegim 2 arxius per a comparar, ens donarà error. NO es pot afegir un número diferent d'arxius.

La comanda principal d'aquest script és **diff**. L'utilitzem en 2 ocasions, per a retornar el número de línies diferent i també per a que ens guardi les línies diferents a error.txt.

Els càlculs els realitzem restant les línies diferents a les línies totals i així extraïem el número de línies correctes. Un cop tenim el número total de línies correctes, realitzem la formula de l'enunciat per a extreure un percentatge de similitud per la sortida estàndard.

2.2 Codi

```
#!/bin/bash
```

```
#Condicional per a que no pugui afegir un número diferent a 2 arxius.
```

```
if [ $# -ne 2 ]; then
```

```
    echo "Has de afegir 2 arxius">&2
```

```
    exit
```

```
fi
```

```
arx1=$1
```

```
arx2=$2
```

```
if [ -f $arx1 ] && [ -f $arx2 ]; then
```

```
    #Contador de línies totals dels arxius.
```

```
    lintot1=$(wc -l $arx1 | cut -d" " -f1)
```

```
    lintot2=$(wc -l $arx2 | cut -d" " -f1)
```

#-l: Sols mostra les línies del fitxer.

#Conta quantes línies tenen errors el primer fitxer.

```
lindifer=$(diff $arx1 $arx2 | grep "<" | wc -l)
```

#Calculem les línies que tenen iguals o bé.

#En el cas de que els documents estan en blanc, significa que els documents són 100% iguals.

#Com després realitzem la multiplicació per 100, llavors percen = 1.

```
if [ $lintot1 -eq 0 ] && [ $lintot2 -eq 0 ]; then
```

```
    percen=1
```

```
else
```

```
    if [ $lintot1 -le $lintot2 ]; then
```

```
        let linbe=$lintot2-$lindifer
```

```
        percen=$(echo "scale=4; $linbe/$lintot2" | bc)
```

```
    else
```

```
        let linbe=$lintot1-$lindifer
```

```
        percen=$(echo "scale=4; $linbe/$lintot1" | bc)
```

```
    fi
```

```
fi
```

```
final=$(echo "$percen*100" | bc)
```

```
echo "$final"
```

#mostra les línies diferents i les passa a error.txt.

```
error=$(diff -y -B -E -b -i --suppress-common-lines $arx1 $arx2 | cat -n)
```

```
echo "$error">&2
```

#-i: Diferencia entre majúscules i minúscules de los ficheros.

#-E: ignora tabuladores.

#-B: Ignora las líneas en blanco.

#-b: comparación omitiendo los espacios en blanco.

#-y mostrar dos columnas para comparar gráficamente.

```
else
```


echo "Han de ser 2 arxius. No pot ser cap tipus diferent o no existeix aquest arxiu">&2

exit

fi

2.3 Joc de proves

Descripció	Característiques	Resultat
Usuari afegeix un número diferent d'arxius a 2.	<code>./compfitxer.sh exemple.txt 2>>error.txt</code>	<u>error.txt</u> Has de afegir 2 arxius.
Usuari afegeix dos directoris o un arxiu no existeix.	<code>./compfitxer.sh exemple.txt cor 2>>error.txt</code>	<u>error.txt</u> Han de ser 2 arxius. No pot ser cap tipus diferent o no existeix aquest arxiu.
Usuari afegeix dos arxius completament blancs.	<code>./compfitxer.sh exemple.txt cp_exemple.txt 2>>error.txt</code>	100.0000
Usuari afegeix correctament 2 arxius amb línies escrites.	<code>./compfitxer.sh exemple.txt cp_exemple.txt 2>>error.txt</code>	<u>estàndard</u> 28.5700 <u>error.txt</u> <i>línies diferents</i>

3. Bash mesactual.sh & Python mesactual.py

3.1 Mesactual.sh

3.1.1 Disseny

En aquest script, hem de tenir en compte que alhora d'executar l'script tot sol, per a no eliminar la informació que ja està guardada al document error.txt hem d'afegir al final 2>>error.txt

En aquest cas, sols podem afegir 2 arxius. En el cas d'afegir un número diferent, tornarà un error pel canal d'error. En el cas de que els dos no siguin arxius, també tornarà un error pel canal d'error.

Les comandes més importants són **stat** i **readlink**. **Stat** ens retorna la data de modificació. **Readlink** ens retorna la ruta absoluta d'un arxiu.

En el cas de que les dues dates siguin exactament igual, ens mostrarà un missatge indicant que les dues dates són iguals.

3.1.2 Codi

```
#!/bin/bash
```

```
#Condicional per a que no pugui afegir un número diferent a 2 arxius.
```

```
if [ $# -ne 2 ]; then
```

```
    echo "Has de afegir 2 arxius">&2
```

```
    exit
```

```
fi
```

```
arx1=$1
```

```
arx2=$2
```

```
#Condicional per a que els fitxers que no siguin arxius, doni error.
```

```
if [ -f $arx1 ] && [ -f $arx2 ]; then
```

```
    #Obtenim la data de modificació actual.
```

```
    datamodi1=$(stat -c %Y $arx1)
```

```
    datamodi2=$(stat -c %Y $arx2)
```

```
    #Realitzem les comparacions.
```

```
    if [ $datamodi1 -gt $datamodi2 ]; then
```

```
        echo "$(readlink -e $arx1)"
```

```

elif [ $datamodi1 -lt $datamodi2 ]; then
    echo "$(readlink -e $arx2)"
else
    echo "$Data de modificació igual"
fi
else
    echo "Han de ser 2 arxius. No pot ser cap tipus diferent o no existeix aquest arxiu">&2
exit
fi

```

3.1.3 Joc de proves

Descripció	Característiques	Resultat
Usuari afegeix un número diferent d'arxius a 2.	<code>./mesactual.sh exemple.txt 2>>error.txt</code>	<u>error.txt</u> Has de afegir 2 arxius.
Usuari afegeix dos directoris o un arxiu no existeix.	<code>./mesactual.sh exemple.txt cor 2>>error.txt</code>	<u>error.txt</u> Han de ser 2 arxius. No pot ser cap tipus diferent o no existeix aquest arxiu.
Usuari afegeix correctament 2 arxius.	<code>./mesactual.sh exemple.txt cp_exemple.txt 2>>error.txt</code>	<u>estàndard</u> <i>ruta absoluta de l'arxiu més recent.</i>

3.2 Mesactual.py

3.2.1 Disseny

En aquest apartat hem decidit importar els paquets **time** i **os**, **time** per a poder saber quin és el fitxer més recentment modificat i el **os** per a poder obtenir la ruta absoluta del fitxer que ha sigut modificat més recentment.

3.2.2 Codi

```

#!/usr/bin/env python3

import os

```

```
import time

#demano els fitxers a l'usuari
arx1= input("Introdueix el primer arxiu: ")
arx2= input("Introdueix el segon arxiu: ")

#miro el temps de la ultima modificacio
actual1 = os.path.getmtime(arx1)
actual2 = os.path.getmtime(arx2)

#comparo quin sa modificat anteriorment + mostro el path absolut
if actual1 > actual2:
    print(os.path.abspath(arx1))
else:
    print(os.path.abspath(arx2))
```

3.2.3 Joc de proves

Descripció	Característiques	Resultat
El programa demana a l'usuari dos fitxers.	Programa: Introdueix el primer arxiu: Usuari: prova1. Introdueix el segon arxiu: Usuari: prova2.	El programa retorn la ruta absoluta del fitxer que ha estat modificat més recentment.
El programa demana a l'usuari dos fitxers.	Programa: Introdueix el primer arxiu: Usuari: prova1.txt Introdueix el segon arxiu: Usuari: prova2.txt	Programa: el programa peta, perquè hem d'afegir arxius sense cap mena d'extensió.
El programa demana a l'usuari dos fitxers. Però l'usuari introdueix dos directoris.	Programa: Introdueix el primer arxiu: Usuari: ex1 (directori). Introdueix el segon arxiu: Usuari: ex2 (directori).	Programa: el programa mostra la ruta absoluta del fitxer que s'ha modificat més recentment de dins d'aquests directoris.

4. Bash compdir.sh

4.1 Disseny

En aquest script, hem de tenir en compte que alhora d'executar l'script tot sol, per a no eliminar la informació que ja està guardada al document error.txt hem d'afegir al final 2>>error.txt

En aquest script hem de tenir en compte que han de ser 2 directoris obligatòriament. El càlcul el tenim en compte segons la quantitat de arxius que hi ha als directoris. Segons quin directori tingui més arxius, significa que més diferencia tenen.

La comanda que més utilitzem és **find**. **Find** s'utilitza per a buscar dins del contingut del fitxer seleccionat. El percentatge es mostra per sortida estàndard i les dates dels arxius més recents es guarda a recents.log.

En el cas de que els dos directoris no tinguin arxius, la seva similitud és del 100%.

4.2 Codi

```
#!/bin/bash
```

```
#Condicional per a que no pugui afegir un número diferent a 2 directoris.
```

```
if [ $# -ne 2 ]; then
```

```
    echo "Has de afegir 2 directoris">&2
```

```
    exit
```

```
fi
```

```
dir1=$1
```

```
dir2=$2
```

```
#Condicional per a comprovar que són 2 directoris i no un tipus diferent.
```

```
if [ -d $dir1 ] && [ -d $dir2 ]; then
```

```
    total=0
```

```
    #Total de arxius dins d'un directori.
```

```
    num1=$(find $dir1 -maxdepth 1 -type f | wc -l)
```

```
    num2=$(find $dir2 -maxdepth 1 -type f | wc -l)
```

```
    #Ruta del primer arxiu.
```

```
    arx1=$(find $dir1 -maxdepth 1 -type f | sort | head -n 1 | tail -n 1)
```



```

    arx2=$(find $dir2 -maxdepth 1 -type f | sort | head -n 1 | tail -n 1)
    #Guardem la ruta absoluta del primer arxiu.
    data1=$(readlink -e $arx1)
    data2=$(readlink -e $arx2)
    #Comparem els dos primers arxius.
    percen=$(./compfitxer.sh $arx1 $arx2 2>>error.txt)
    #Anem guardant tots els percentatges dels arxius dins de total.
    total=$(echo "$total+$percen" | bc)

    # Condicional per a veure si els dos directoris tenen 0 arxius.
    if [ $num1 -eq 0 ] && [ $num2 -eq 0 ]; then
        #En el cas de no tenir cap arxiu, els directoris son exactament iguals.
        final=100
    else
        #Condicional per a veure si el primer directori te menys arxius que el segon
        if [ $num1 -le $num2 ]; then

            #Bucle per a anar comparant arxiu per arxiu.
            #Sols tindrem en compte el número de arxius del directori amb menys
arxius.

            for (( i=2; i<=$num1; i++ ))
            do
                #Anem agafant arxius
                arx1=$(find $dir1 -maxdepth 1 -type f | sort | head -n $i | tail -n 1)
                arx2=$(find $dir2 -maxdepth 1 -type f | sort | head -n $i | tail -n 1)
                data1=$(./mesactual.sh $arx1 $data1 2>>error.txt)
                percen=$(./compfitxer.sh $arx1 $arx2 2>>error.txt)
                total=$(echo "$total+$percen" | bc)
            done
            #El percentatge de similitud hem de tenir en compte el directori amb més
arxius.

            final=$(echo "scale=2; $total/$num2" | bc)

```



analitzat #Tenim en compte les dates de modificació dels arxius que no s'han

```
arx2=$(find $dir2 -maxdepth 1 -type f | sort | head -n 1 | tail -n 1)
data2=$arx2
#Bucle per a veure les dates dels arxius no comparats.
for (( i=2; i<=$num2; i++ ))
do
    arx2=$(find $dir2 -maxdepth 1 -type f | sort | head -n $i | tail -n 1)
    data2=$(./mesactual.sh $arx2 $data2 2>>error.txt)
done
#Guardem les dates al fitxer recents.log
echo "$data1" >> recents.log
echo "$data2" >> recents.log
```

#Si el segon directori te menys arxius que el primer
else

```
#Bucle per a anar comparant arxiu per arxiu
for (( i=2; i<=$num2; i++ ))
do
    arx1=$(find $dir1 -maxdepth 1 -type f | sort | head -n $i | tail -n 1)
    arx2=$(find $dir2 -maxdepth 1 -type f | sort | head -n $i | tail -n 1)
    data1=$(./mesactual.sh $arx1 $data1 2>>error.txt)
    percen=$(./compfitxer.sh $arx1 $arx2 2>>error.txt)
    total=$(echo "$total+$percen" | bc)
done
```

els> #Per a saber el percentatge s'ha de dividir pel numero maxim de arxius entre

```
final=$(echo "scale=2; $total/$num1" | bc)
#Tenim en compte les dates de modificacions dels arxius no analitzats
arx1=$(find $dir1 -maxdepth 1 -type f | sort | head -n 1 | tail -n 1)
data1=$arx1
#Bucle per a anar veent els arxius que no hem analitzat.
```

```

for (( i=2; i<=$num1; i++ ))
do
    arx1=$(find $dir1 -maxdepth 1 -type f | sort | head -n $i | tail -n 1)
    data1=$(./mesactual.sh $arx1 $data1 2>> error.txt)
done
#Guardem les rutes absolutes a recents.log.
echo "$data1" >> recents.log
echo "$data2" >> recents.log
fi
fi
#Mostrem el resultat final per sortida estàndard.
echo "$final"
else
    echo "Han de ser 2 directoris. No pot ser cap tipus diferent o el directori ha de
existir">&2
    exit
fi

```

4.3 Joc de proves

Descripció	Característiques	Resultat
Usuari afegeix un número diferent d'arxius a 2.	./compdir.sh ./compdir.sh comparacio_compdir/directori_comp aracio/ 2>>error.txt	<u>error.txt</u> Has de afegir 2 directoris.
Usuari afegeix dos directoris o un directori no existeix.	./compdir.sh comparacio_compdir/directori_comp aracio/ cor 2>error.txt	<u>error.txt</u> Han de ser 2 directoris. No pot ser cap tipus diferent o el directori ha de existir.
Usuari afegeix correctament 2 directoris.	./compdir.sh comparacio_compdir/directori_comp aracio/ comparacio_compdir/cp_directori_c omparacio/ 2>error.txt	<u>estàndard</u> <i>percentatge de similitud.</i> <u>recents.log</u> <i>ruta d'arxius més recents.</i>

5. Bash comptot.sh & Python comptot.py

5.1 Comptot.sh

5.1.1 Disseny

En aquest script, hem de tenir en compte que alhora d'executar l'script tot sol, per a no eliminar la informació que ja està guardada al document error.txt hem d'afegir al final 2>>error.txt

En aquest script tenim en compte el nombre de subdirectoris i el nombre d'arxius totals del directori (també tenint en compte els subarxius) per a calcular el percentatge de similitud. Es tenen en compte el total més gran de arxius totals i el total més gran de subdirectoris totals.

Realitzem una funció casi igual que el script compdir però recorrent tots els subdirectoris i els seus subarxius.

En aquest script les comandes més utilitzades són un conjunt utilitzades anteriorment.

5.1.2 Codi

```
#!/bin/bash
```

```
#Condicional per a que no pugui afegir un número diferent a 2 directoris.
```

```
if [ $# -ne 2 ]; then
```

```
    echo "Has de afegir 2 directoris">&2
```

```
    exit
```

```
fi
```

```
dir1=$1
```

```
dir2=$2
```

```
#Condicional per a comprovar que són 2 directoris i no un tipus diferent.
```

```
if [ -d $dir1 ] && [ -d $dir2 ]; then
```

```
    tot=0
```

```
    #Numero de directoris total
```

```
    numdir1=$(ls -F $dir1 | grep / | wc -l)
```

```
    numdir2=$(ls -F $dir2 | grep / | wc -l)
```

```
    #Numero total de arxius nomes en el directori actual
```

```
    numarx1=$(find $dir1 -maxdepth 1 -type f | wc -l)
```

```

numarx2=$(find $dir2 -maxdepth 1 -type f | wc -l)
#Numero total de arxius, contant tambe els subarxius
numarxtot1=$(find $dir1 -type f | wc -l)
numarxtot2=$(find $dir2 -type f | wc -l)
#Condicional per si els dos directoris no tenen subdirectoris
if [ $numdir1 -eq 0 ] && [ $numdir2 -eq 0 ]; then
    final=$(./compdir.sh $dir1 $dir2 2>>error.txt)
#Si no es compleix significa que minim 1 dels 2 directoris te subdirectoris
else
    #Si els primer directori no te cap subdirector
    if [ $numdir1 -eq 0 ]; then
        #Si el numero de arxius total (tenin en compte tambe els subdirectoris) es
mes p>
        if [ $numarxtot2 -le $numarx1 ]; then
            final=$(./compdir.sh $dir1 $dir2 2>>error.txt)
            #Si el numero d'arxius del directori 1 es mes petit que que el total de arxius
d>
        else
            percen=$(./compdir.sh $dir1 $dir2 2>>error.txt)
            total=$(echo "scale = 2; $percen*$numarx1" | bc)
            final=$(echo "$total/$numarxtot2" | bc)
        fi
    #Bucle per a comparar les dates dels arxius del subdirector
    for (( i=2; i<=$numdir2+1; i++ ))
    do
        subdir2=$(find $dir2 -maxdepth 1 -type d | sort | head -n $i | tail -n 1)
        numsubarx2=$(find $subdir2 -maxdepth 1 -type f | wc -l)
        subarx2=$(find $subdir2 -maxdepth 1 -type f | sort | head -n 1 | tail -n>
        data2=$(readlink -e $subarx2)
        for (( j=2; j<=$numsubarx2; j++ ))
        do
            subarx2=$(find $subdir2 -maxdepth 1 -type f | sort | head -n $j >

```




```
data2=$(./mesactual.sh $subarx2 $data2 2>>error.txt)
done
echo "$data2" >> recents.log
done
#Si els segon directori no te cap subdirectori i el primer si
elif [ $numdir2 -eq 0 ]; then
    #Si el numero de arxius total del directori 1 es mes petit que el directori 1
    if [ $numarxtot1 -le $numarx2 ]; then
        final=$(./compdir.sh $dir1 $dir2 2>>error.txt)
        #Si el numero de arxius del directori 1 es mes petit que el total de arxius del
>
    else
        percen=$(./compdir.sh $dir1 $dir2 2>>error.txt)
        total=$(echo "scale = 2; $percen*$numarx2" | bc)
        final=$(echo "$total/$numarxtot1" | bc)
    fi
    #Bucle per a comparar les dates dels arxius del subdirectori
    for (( i=2; i<=$numdir1+1; i++ ))
    do
        subdir1=$(find $dir1 -maxdepth 1 -type d | sort | head -n $i | tail -n 1)
        numsubarx1=$(find $subdir1 -maxdepth 1 -type f | wc -l)
        subarx1=$(find $subdir1 -maxdepth 1 -type f | sort | head -n 1 | tail -n>
        data1=$(readlink -e $subarx1)
        for (( j=2; j<=$numsubarx1; j++ ))
        do
            subarx1=$(find $subdir1 -maxdepth 1 -type f | sort | head -n $j >
            data1=$(./mesactual.sh $subarx1 $data1 2>>error.txt)
        done
        echo "$data1" >> recents.log
    done
    #Si els dos directoris tenen com a minim 1 sudirectori
else
```



#Primer comparem únicament els arxius del directori pare

if [\$numarx1 -le \$numarx2]; then

 percen1=\$(./compdir.sh \$dir1 \$dir2 2>>error.txt)

 total1=\$(echo "scale = 2; \$percen1*\$numarx2" | bc)

else

 percen1=\$(./compdir.sh \$dir1 \$dir2 2>>error.txt)

 total1=\$(echo "scale = 2; \$percen1*\$numarx1" | bc)

fi

#Comencem a analitzar els subdirectoris

subdir1=\$(find \$dir1 -maxdepth 1 -type d | sort | head -n 2 | tail -n 1)

subdir2=\$(find \$dir2 -maxdepth 1 -type d | sort | head -n 2 | tail -n 1)

numsubarx1=\$(find \$subdir1 -maxdepth 1 -type f | wc -l)

numsubarx2=\$(find \$subdir2 -maxdepth 1 -type f | wc -l)

#Comparem els subarxius

if [\$numsubarx1 -le \$numsubarx2]; then

 percen2=\$(./compdir.sh \$subdir1 \$subdir2 2>>error.txt)

 total2=\$(echo "scale = 2; \$percen2*\$numsubarx2" | bc)

else

 percen2=\$(./compdir.sh \$subdir1 \$subdir2 2>>error.txt)

 total2=\$(echo "scale = 2; \$percen2*\$numsubarx1" | bc)

fi

totalfinal=\$(echo "\$total1+\$total2" | bc)

#El total que tenim el dividim per el total maxim de arxius contan els del

subdi>

#El directori que tingui mes, llavors utilitzarem el seu valor per a la divisio

if [\$numarxtot1 -le \$numarxtot2]; then

 final1=\$(echo "scale = 2; \$totalfinal/\$numarxtot2" | bc)

else

 final1=\$(echo "scale = 2; \$totalfinal/\$numarxtot1" | bc)

fi

tot=\$(echo "\$tot+\$final1" | bc)

#Aquesta condicio seria per si tenim un numero de subdirectoris diferent



```
if [ $numdir1 -le $numdir2 ]; then
    for (( i=3; i<=$numdir1+1; i++ ))
    do
        subdir1=$(find $dir1 -maxdepth 1 -type d | sort | head -n $i | t>
        subdir2=$(find $dir2 -maxdepth 1 -type d | sort | head -n $i | t>
        numsubarx1=$(find $subdir1 -maxdepth 1 -type f | wc -l)
        numsubarx2=$(find $subdir2 -maxdepth 1 -type f | wc -l)
        if [ $numsubarx1 -le $numsubarx2 ]; then
            percen2=$(./compdir.sh $subdir1 $subdir2 2>>error.txt)
            total2=$(echo "scale = 2; $percen2*$numsubarx2" | bc)
        else
            percen2=$(./compdir.sh $subdir1 $subdir2 2>>error.txt)
            total2=$(echo "scale = 2; $percen2*$numsubarx1" | bc)
        fi
        totalfinal=$(echo "$total1+$total2" | bc)
        if [ $numarxtot1 -le $numarxtot2 ]; then
            final1=$(echo "scale = 2; $totalfinal/$numarxtot2" | bc)
        else
            final1=$(echo "scale = 2; $totalfinal/$numarxtot1" | bc)
        fi
        tot=$(echo "$tot+$final1" | bc)
    done
    final=$(echo "scale=2; $tot/$numdir2" | bc)
    #Bucle per a comparar les dates del subarxiu no comparat
    for (( i=$numdir1+1; i<=$numdir2+1; i++ ))
    do
        subdir2=$(find $dir2 -maxdepth 1 -type d | sort | head -n $i | t>
        numsubarx2=$(find $subdir2 -maxdepth 1 -type f | wc -l)
        subarx2=$(find $subdir2 -maxdepth 1 -type f | sort | head -n 1 |>
        data2=$(readlink -e $subarx2)
        for (( j=2; j<=$numsubarx2; j++ ))
```



```
do
    subarx2=$(find $subdir2 -maxdepth 1 -type f | sort | head -n 1)
    data2=$(./mesactual.sh $subarx2 $data2 2>>error.txt)
done
echo "$data2" >> recents.log
done
else
for (( i=3; i<=$numdir2+1; i++ ))
do
    subdir1=$(find $dir1 -maxdepth 1 -type d | sort | head -n $i | tail -n 1)
    subdir2=$(find $dir2 -maxdepth 1 -type d | sort | head -n $i | tail -n 1)
    numsubarx1=$(find $subdir1 -maxdepth 1 -type f | wc -l)
    numsubarx2=$(find $subdir2 -maxdepth 1 -type f | wc -l)
    if [ $numsubarx1 -le $numsubarx2 ]; then
        percen2=$(./compdir.sh $subdir1 $subdir2 2>>error.txt)
        total2=$(echo "scale = 2; $percen2*$numsubarx2" | bc)
    else
        percen2=$(./compdir.sh $subdir1 $subdir2 2>>error.txt)
        total2=$(echo "scale = 2; $percen2*$numsubarx1" | bc)
    fi
    totalfinal=$(echo "$total1+$total2" | bc)
    if [ $numarxtot1 -le $numarxtot2 ]; then
        final1=$(echo "scale = 2; $totalfinal/$numarxtot2" | bc)
    else
        final1=$(echo "scale = 2; $totalfinal/$numarxtot1" | bc)
    fi
    tot=$(echo "$tot+$final1" | bc)
done
final=$(echo "scale=2; $tot/$numdir1" | bc)
for (( i=$numdir2+1; i<=$numdir1+1; i++ ))
do
```

```

subdir1=$(find $dir1 -maxdepth 1 -type d | sort | head -n $i | t>
numsubarx1=$(find $subdir1 -maxdepth 1 -type f | wc -l)
subarx1=$(find $subdir1 -maxdepth 1 -type f | sort | head -n 1 |>
data1=$(readlink -e $subarx1)
for (( j=2; j<=$numsubarx1; j++ ))
do
    subarx1=$(find $subdir1 -maxdepth 1 -type f | sort | hea>
    data1=$(./mesactual.sh $subarx1 $data1 2>>error.txt)
done
echo "$data1" >> recents.log
done
fi
fi
fi
echo "$final"
else
    echo "Han de ser 2 directoris. No pot ser cap tipus diferent o el directori ha de
existir">&2
    exit
fi

```

5.1.3 Joc de proves

Descripció	Característiques	Resultat
Usuari afegeix un número diferent d'arxius a 2.	./comptot.sh comparacio_comptot/comp1 2>>error.txt	<u>error.txt</u> Has de afegir 2 directoris.
Usuari afegeix dos directoris o un directori no existeix.	./comptot.sh comparacio_comptot/comp1 cor 2>>error.txt	<u>error.txt</u> Han de ser 2 directoris. No pot ser cap tipus diferent o el directori ha de existir.
Usuari afegeix correctament 2 directoris.	./comptot.sh comparacio_comptot/comp1 comparacio_comptot/comp2 / 2>>error.txt	<u>estàndard</u> <i>percentatge de similitud.</i> <u>recents.log</u> <i>ruta d'arxius més recents.</i>

5.2 Comptot.py

5.2.1 Disseny

En aquest apartat l'hauria d'haver implementat, però malauradament per la falta de temps y de coneixements per a programar en Python no he pogut completar aquest apartat, la funció d'aquest comptot.py seria per comparar el contingut de dos directoris, el primer dir1 i el segon dir2 en el qual en cada un d'aquests directoris haguéssim tingut diversos subdirectoris i dins d'aquest subdir haguéssim tingut fitxer, la funcionalitat seria comparar tots els fitxers a escala de directori principal i de subdirectori i d'aquesta manera obtenir un percentatge de similitud entre totes aquestes comparacions.

El que he pogut aconseguir és el percentatge de similitud a escala dels directoris, gràcies a la comanda subprocess.run (hem invertit moltíssim de temps per saber com funcionava, ja que sense aquesta comanda no haguéssim pogut fer res), seguidament he implementat un bucle per a saber quants subdirectoris tenim dins les carpetes, un bucle per a saber quants fitxers tenim en el directori, i un últim bucle per a saber quants fitxers tenim en total (contant els fitxers del mateix directori i del subdirectori).

Seguidament, he fet diversos càlculs de percentatge que m'aguissin sigut molt útils més endavant, però no he pogut invertir més temps, i m'he quedat en l'apartat de fer un bucle per a comparar les dates dels arxius del subdirectori i d'aquesta manera obtenir quin és el més recent.

5.2.2 Codi

```
#!/usr/bin/env python3

import os
import filecmp
import subprocess
import sys

from subprocess import Popen, PIPE

import pathlib

tot = 0
total = 0

#ESTO FUNCIONA CON FICHEROS

#subprocess.Popen("./mesactual.py prova1 prova2", shell=True)

arx1= input("Introdueix el primer directori: ")
arx2= input("Introdueix el segon directori: ")

#NUMERO DE DIRECTORIOS
```

```
numdir1 = 0
numdir2 = 0
dir1 = arx1
dir2 = arx2
#NUMERO DE DIRECTORIOS
for path in os.listdir(dir1):
    if os.path.isdir(os.path.join(dir1,path)):
        numdir1 += 1
print("Numero del primer directori:",numdir1)
for path in os.listdir(dir2):
    if os.path.isdir(os.path.join(dir2,path)):
        numdir2 += 1
print("Numero del segon directori:",numdir2)
numarx1 = 0
numarx2 = 0
#NUMERO DE FITXERS EN EL PRIMER DIRECTORI
for path in os.listdir(dir1):
    if os.path.isfile(os.path.join(dir1,path)):
        numarx1 += 1
print("Numero total dels arxius del primer dir:",numarx1)
contador=0
#NUMERO DE FITXERS EN EL SEGON DIRECTORI
for path in os.listdir(dir2):
    if os.path.isfile(os.path.join(dir2,path)):
        numarx2 += 1
print("Numero total dels arxius del segon dir:", numarx2)
numarxtot1 = 0
numarxtot2 = 0
#NUMERO TOTAL DE FITXERS, CONTAN ELS SUBARXIUS
for path, subdirs, files in os.walk(dir1):
    for name in files:
```

```

numarxtot1 += 1
print("Numero total de fitxers de TOT el 1r directori:",numarxtot1)
for path, subdirs, files in os.walk(dir2):
    for name in files:
        numarxtot2 += 1
print("Numero total de fitxer de TOT el 2n directori:",numarxtot2)

if numdir1 == 0 and numdir2 == 0:
    final = subprocess.run(["./compdir.sh {x} {y}".format(x=arx1,
y=arx2)],capture_output=True, text=True,shell=True)
    print("NO FUNCIONA")
else:
    if numdir1 == 0:
        if numarxtot2 <= numarxtot1:
            final = subprocess.run(["./compdir.sh {x} {y}".format(x=arx1,
y=arx2)],capture_output=True, text=True,shell=True)
            print("MES GRAN QUEL DIR2: ")
            #FINAL IF
        else:
            tot = subprocess.run(["./compdir.sh {x} {y}".format(x=arx1,
y=arx2)],capture_output=True, text=True,shell=True)
            fin = tot.stdout
            fin=fin.strip('\n')
            percen=float(fin)
            print("PERCENTATJE CALCULAT(dir1 dir2): ",percen)
            def multiplicar(per,num1): #DEF PER A MULTIPLICAR
                return per*num1
            total = multiplicar(percen,numarx1)
            total = percen * numarx1
            print("CALCUL TOTAL(perc*numarx1): ",total)
            def dividir(num,div):
                return num/div

```

```

try: #TRY ES PER A DIVIDIR
    final = dividir(total, numarxtot2)
    print("CALCUL FINAL(total/numarxtot2): ",final)
except ZeroDivisionError:
    print("PERRO NO POTS DIVIDIR PER 0")
#FINAL ELSE
i = 2
#COMPAREM DATES DELS ARXIS DEL SUBDIRECTORI
numdir2 = numdir2+1
while i <= numdir2:
    #for i in range(2, numdir2):
        print("ENTRO AL FORRRRR")
        #me llista tots los subfixers i fixers
        def busc_fit(dir2):
            for fitx2 in os.listdir(dir2):
                full_path = os.path.join(dir2,fitx2)
                if os.path.isdir(full_path):
                    busc_fit(full_path)
                else:
                    #-----M'HE QUEDAT EN AQUEST APARTAT -----
                    if os.path.isfile(full_path):
                        print("FICHEROS: ",os.path.join(dir,full_path))
                        #filecmp.cmpfiles(os.path.getmtime(full_path))
                    busc_fit(dir2)
            #totdir2 = os.path.isfile(dir2)
            #print(totdir2)
            #subdir2 = os.path.getmtime(dir2)
            #numsubarx2 = os.path.getmtime(subdir2)
            #subarx2 = os.path.getmtime(subdir2)
            #data2 = os.path.realpath(subarx2)
            #

```

```
#contingut = os.listdir(dir2)
#totdir2= os.listdir(dir2)
#print(totdir2)
i = i + 1
j = 2
while j <= numsubarx2:
#for j in range(2, numsubarx2):
    subarx2 = os.path.getmtime(subdir2)
    data2 = subprocess.Popen("./mesactual.sh prova1 prova2", shell=True)
    j = j + 1
    print("FITXER MODIFICAT RECENMENT: ",data2)>>recents.log
#FINAL DEL FOR
#SI ELS SEGON DIRECTORI NO TE CAP SUBDIRECTORI I EL PRIMER SI
elif numdir2 == 0:
    if numarxtot1 <= numarx2:
        #final = subprocess.run(["./compdir.sh {x} {y}".format(x=arx1,
y=arx2)],capture_output=True, text=True,shell=True)
        print("hola")
    else:
        #percen = subprocess.run(["./compdir.sh {x} {y}".format(x=arx1,
y=arx2)],capture_output=True, text=True,shell=True)
        print("h")
    i = 2
    for i in range(2, numdir1+1):
        subdir1 = os.walk(dir1)
        numsubarx1 = os.walk(subdir1)
        subarx1 = os.walk(subdir1)
        data1 = subarx1
        j = 2
        for j in range(2, numsubarx1):
            subarx1 = os.walk(subdir1)
            #data1 = subprocess.Popen("./mesactual.py prova1 prova2", shell=True)
```



```
#print(data1)>>recents.log

#FI FOR

#FI ELIF

else:

    print("ARRIBO?")

print("HASTA AQUI")
```

5.2.3 Joc de proves

Descripció	Característiques	Resultat
<p>El programa demana a l'usuari introduir dos directoris.</p> <p>El programa conta el nombre de subdirectoris que tenen cada directori de forma independent, també conta el nombre d'arxius que tenen al directori principal i també el nombre total de fitxers i subfitxers.</p> <p>Calcula el percentatge de similitud dels fitxers, i també realitza el percentatge total (percentatge*númers d'arxius del primer directori) i el càlcul final de percentatge (total/nombre d'arxius totals del segon directori).</p>	<p>Programa: Introdueix el primer directori.</p> <p>Usuari: ex1.</p> <p>Programa: Introdueix el segon directori</p> <p>Usuari: ex2</p>	<p>Programa:</p> <p>Nombre del primer directori: 0</p> <p>Nombre del segon directori: 1</p> <p>Nombre total dels arxius del primer dir: 2</p> <p>Nombre total dels arxius del segon dir: 3</p> <p>Nombre total de fitxers de TOT el 1r directori: 2</p> <p>Nombre total de fitxers de TOT el 2n directori: 5</p> <p>PERCENTATGE</p> <p>CÀLCULAT (dir1 dir2): 66.66</p> <p>CÀLCUL TOTAL</p> <p>(perc*numarx1): 133.32</p> <p>CÀLCUL FINAL</p> <p>(total/numarxtot2): 26.663999999999998</p>

6. Bash creanou.sh

6.1 Disseny

En aquesta pràctica l'usuari ha d'introduir un directori, i aquest programa crearà dins d'un directori rebut per paràmetre, els fitxers i directoris necessaris, mantenint les dates de creació i modificació dels diferents fitxers.

Primerament, verifiquem que l'usuari ha introduït un directori i crearem una carpeta per si no existeix la que l'usuari ha introduït, seguidament contem les línies de recents.log i per últim fem un bucle perquè valgué copiant les línies del recent.log.

6.2 Codi

```
#!/bin/bash
```

```
#Condicional per a que no pugui afegir un número diferent a 1 directoris
```

```
if [ $# -ne 1 ]; then
```

```
    echo "Has de afegir 1 directori">&2
```

```
    exit
```

```
fi
```

```
#Demanem el directori on volem copiar els arxius de recent.log
```

```
dir2=$1
```

```
#Crearem la carpeta per si no existeix
```

```
#Contador de quantes línies te recents.log
```

```
num=$(wc -l recents.log | cut -d" " -f1)
```

```
#Bucle per a que vaig copiant de línia en línia el recents.log
```

```
for (( i=1; i<=$num; i++ ))
```

```
do
```

```
    dir1=$(cat recents.log | head -n $i | tail -n 1)
```

```
    #Comanda per copiar arxiu amb data de modificació
```

```
    rsync -rv --times $dir1 $dir2
```

```
done
```

6.3 Joc de proves

Descripció	Característiques	Resultat
El programa demana a l'usuari d'introduir un directori.	Usuari: ex1.	Programa: copia les línies del recent.log i també copia arxius amb data de modificació.
El programa demana a l'usuari d'introduir un directori. L'usuari introdueix un directori que no existeix.	Usuari: e.	Programa: crea un directori anomenat dir2 i copia les línies del recent.log i també copia arxius amb data de modificació.

7. Bash p1.sh & Python p1.py

7.1 P1.sh

7.1.1 Disseny

En aquest script sols tenim que tenir una recopilació i una crida als altres scripts per a que funcionin correctament.

Hem de tenir en compte que el ultim directori ha d'existir i ha d'estar creat ja que si no esta creat, l'script no funcionarà correctament i donarà un missatge d'error.

7.1.2 Codi

```
#!/bin/bash
```

```
#Condicional per a que no pugui afegir un número diferent a 3 fitxers.
```

```
if [ $# -ne 3 ]; then
```

```
    echo "Has de afegir 3 fitxers">&2
```

```
    exit
```

```
fi
```

```
fit1=$1
```

```
fit2=$2
```

```
dir=$3
```

```
#Condicional per a que el els dos primer fitxers siguin arxius i l'ultim un directori
```

```
if [ -f $fit1 ] && [ -f $fit2 ] && [ -d $dir ]; then
```

```
    ./compfitxer.sh $fit1 $fit2 2>>error.txt
```

```
    ./mesactual.sh $fit1 $fit2 >>recents.log 2>>error.txt
```

```
    ./creanou.sh $dir 2>>error.txt
```

```
#Condicional per a que tres siguin directoris
```

```
elif [ -d $fit1 ] && [ -d $fit2 ] && [ -d $dir ]; then
```

```
    ./comptot.sh $fit1 $fit2 2>>error.txt
```

```
    ./creanou.sh $dir 2>>error.txt
```

```
#Condicional en cas de no complir qualsevol dels anteriors
```

```
else
```

echo "Han de ser 3 directoris o 2 arxius i 1 directori. No pot ser cap tipus diferent o el direc>

exit

fi

7.1.3 Joc de proves

Descripció	Característiques	Resultat
Usuari afegeix un numero incorrecte de fitxers	<code>./p1.sh copieta.txt</code>	Has de afegir 3 fitxers
Usuari afegeix dos fitxers de tipus diferent i l'últim tampoc no es un directori o no existeix	<code>./p1.sh comparacio_comptot/comp1 / comparacio_comptot/comp2 .txt copieta.txt</code>	Han de ser 3 directoris o 2 arxius i 1 directori. No pot ser cap tipus diferent o el directoris/arxius han de existir
Usuari afegeix 2 arxius i 1 directori	<code>./p1.sh exemple.txt cp_exemple.txt copieta/</code>	<u>error.txt</u> <i>linies diferents dels arxius</i> <u>recents.log</u> <i>arxiu més recents</i> <u>copieta</u> <i>Es copiaran tots els arxius de recent.log</i>
Usuari afegeix 3 directoris	<code>./p1.sh comparacio_comptot/comp1 / comparacio_comptot/comp2 / copieta</code>	<u>error.txt</u> <i>linies diferents dels arxius i subarxius</i> <u>recents.log</u> <i>arxius i subarxius més recents</i> <u>copieta</u> <i>Es copiaran tots els arxius i subarxius de recent.log</i>

7.2 P1.py

7.2.1 Disseny

En aquest apartat, com ja he explicat anteriorment, no he pogut acabar de completar tot el codi comptot.py malauradament per la falta de coneixements i de temps, però, en canvi, he decidit fer un últim esforç i en aquest apartat mostro el percentatge de similitud per la sortida estàndard dels fitxers d'una comparació de dos directoris, i també he aconseguit afegir la ruta absoluta d'ells fitxers amb dada de modificació més nova per un fitxer.log.

7.2.2 Codi

```
#!/usr/bin/env python3
```

```
import os
```

```
import subprocess
```

```
import sys
```

```
from subprocess import Popen, PIPE
```

```
import pathlib
```

```
#UN PERCENTATGE DE SIMILITUD PER LA SORTIDA ESTANDARD
```

```
arx1= input("Introdueix el primer directori: ")
```

```
arx2= input("Introdueix el segon directori: ")
```

```
tot=subprocess.run(["./compdir.sh {x} {y}".format(x=arx1, y=arx2)],capture_output=True,  
text=True,shell=True)
```

```
percen=tot.stdout
```

```
percen=percen.strip('\n')
```

```
fin=float(percen)
```

```
print(fin)
```

#LA RUTA ABSOLUTA DELS FITXERS AMB DADA DE MODIFICACIO MES NOVA PER UN FITXER.log

```
tot=subprocess.run(["./mesactual.sh {x} {y}"].format(x=arx1,y=arx2)],capture_output=True,
text=True, shell=True)
```

```
percen=tot.stdout
```

```
percen=percen.strip("\n")
```

```
import contextlib
```

```
with open("fitxer.log","w") as o:
```

```
    o.write(percen)
```

7.3.3 Joc de proves

Descripció	Característiques	Resultat
L'usuari introdueix el nom de dos directoris	Programa: Introdueix el primer directori: Usuari: ex1 Introdueix el segon directori: Usuari: ex2	Programa: 66.66 I ens genera un fitxer.log en el qual ens sorgeix la ruta absoluta del fitxer modificat més recentment, cada vegada que s'executa el programa la ruta se sobreescriu.
L'usuari introdueix un directori que no existeix	Programa: Introdueix el primer directori: Usuari: ex1 Introdueix el segon directori: Usuari: a	Programa: el programa peta

Conclusió

L'apartat que més ens ha costat de tots els arxius ha sigut sobretot **comptot.sh** i **comptot.py**. Alhora de tenir en compte totes les possibilitats que poden tenir els directoris i els infinits subdirectoris, no sabiem per on començar. Finalment, vam aclarir uns requisits per a poder solucionar aquest problema.

Un dels problemes més grans que vam tenir era que utilitzaven una versió antiga de python i llavors moltes funcionalitats de les actuals ens donava error al no tenir python actualitzat. En va portar un temps entendre que era aquest el problema. Al actualitzar ja vam poder solucionar el problema i tot funciona correctament.

Finalment, també uns dels problemes va ser que l'enunciat no especificava molt i en alguns punts podia ser molt confús. Al poder elegir nosaltres les especificacions i poder-los explicar a la documentació per a que l'altra gent la entengui, vam poder solucionar aquest problema.