

MEMORIA FONAMENTS DE SISTEMES OPERATIUS

Pràctica 1

Jan Torres Rodríguez

Raúl Martín Morales

Index:

1. Còpia de seguretat

| | |
|----------------------|---|
| 1.1. Disseny | 3 |
| 1.2. Codi del script | 4 |
| 1.3. Joc de proves | 6 |

2. Recuperació d'un directori o fitxer a un altre directori

| | |
|----------------------|----|
| 2.1. Disseny | 8 |
| 2.2. Codi del script | 8 |
| 2.3. Joc de proves | 10 |

3. Automatització amb crontab

| | |
|----------------------|----|
| 3.1. Disseny | 12 |
| 3.2. Codi del script | 12 |
| 3.3. Joc de proves | 13 |

1. Còpia de seguretat

Copia [-K] [-D] fitxer1 fitxer2 ... fitxer_tgz

1.1. Disseny

Per dissenyar el nostre script de copia de seguretat primerament hem decidit fer diverses comprovacions que s'encarregaran de fer que l'usuari escrigui be els arguments correctes sobre el script. Amb l'ús d'un condicional hem comprovat que s'escriguin més de 2 arguments ja que és necessari que almenys s'escrigui un argument per l'arxiu que es vol comprimir i un altre argument per indicar a a quin comprimit es vol fer al còpia.

Després comprovem si en un dels dos primers arguments es troba un "-D", ja que si es troba, l'script haurà de mostrar per pantalla el seu propi codi (Dry Run) amb l'ús de la comanda tail.

Si no ha introduït el "-D" , aleshores mirem si al primer argument s'ha indicat el "-K". Si l'ha indicat utilitzem un flag propi per a determinar l'existència d'aquesta opció posteriorment en altres fases del codi.

El codi a partir d'aquí ja comença a fer la còpia de seguretat. Primer creem un directori temporal on anirem guardant els arxius que es volen comprimir temporalment. Posteriorment, es comprimirà el contringut d'aquest directori i s'eliminarà per a que no en quedi rastre.

Si el comprimit on es volen destinar els arxius ja existeix, afegirem els arxius especificats, però si no existeix crearem el comprimit nosaltres amb la comanda tar.

Seguidament entrem dins d'aquest directori temporal amb la comanda cd, i recorrem els paràmetres amb la comanda for. Mentre no sigui el comprimit ni cap opció d'especificarió, s'introduiran els arxius especificats al directori temporal. Dins d'aquest bucle amb un condicional es mira si la flag del "-K" està activada, si ho està s'empra la comanda "stat" per a obtenir informació sobre la data de la última modificació, per a afegir-la més tard al nom del duplicat.

Si no està activada, mirem si l'arxiu del paràmetre és un directori, si ho és s'executa la comanda cp recursiva, sinó fem la copia normal.

Finalment comprimim tot el que hi ha dins del directori temporal en el comprimit passat per l'últim paràmetre, i eliminem aquesta carpeta temporal. Per acabar, es mostra un missatge indicant que s'ha fet la còpia de seguretat correctament.

1.2. Codi de l'script

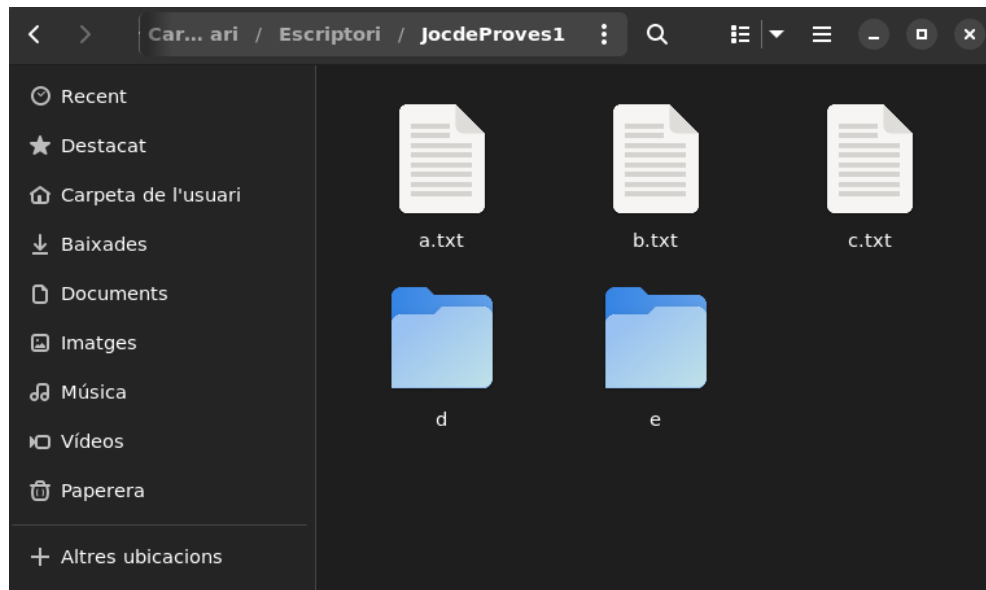
```
let comptador=1
comprimit="$ {@: -1}"
#i serà el primer dels fitxers a comprimir
if [ $# -lt 2 ]
then
echo "Necessites mes arguments"
exit 1
fi

if [ "$1" = "-D" ] || [ "$2" = "-D" ]
then
tail -39 $0
exit 0
else
if [ "$1" = "-K" ]
#S'utilitza i com a flag del primer arxiu
then let i=2
else let i=1
fi
#Copia de seguretat: compressio
temporal="dirTemp"
mkdir "$temporal"
if [ -e $comprimit ]
then tar -xpf "$comprimit" -C "$temporal"
fi
cd "$temporal"
for arxiu in $*
do
if [ $comptador -ge $i ] && [ $comptador -ne $# ]
then
```

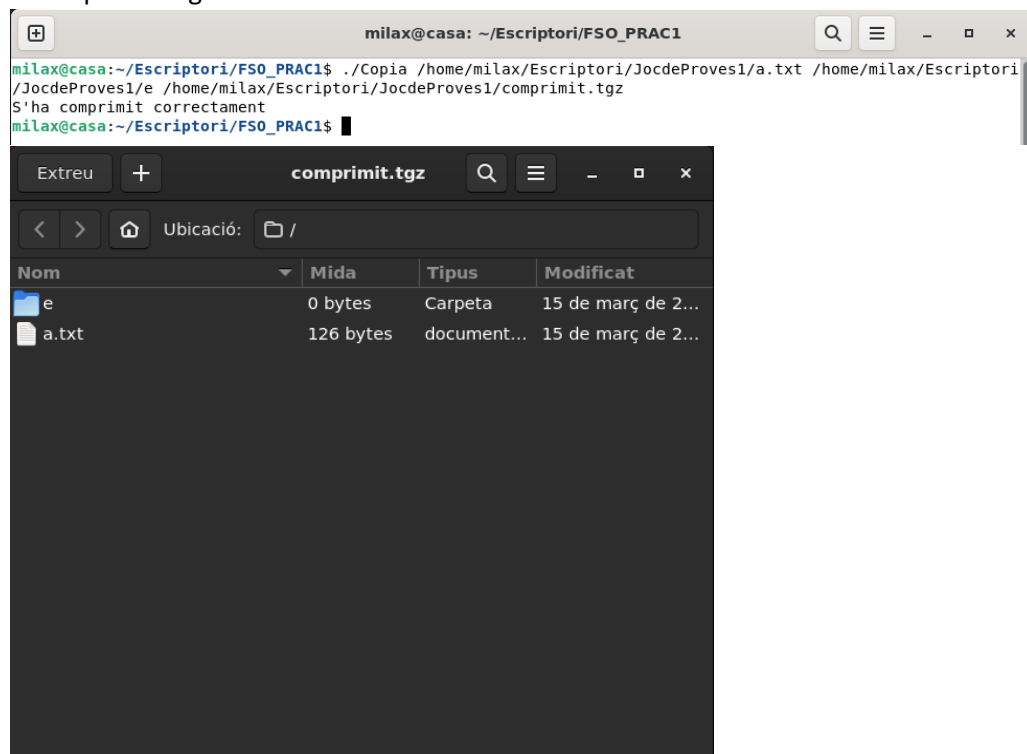
```
if [ $i -eq 2 ]
then
data=$(stat -c %y "$arxiu" | cut -d' ' -f1 | sed 's/-//g')
nomNou="$arxiu.$data"
if [ -d $arxiu ]
then
cp -r "$arxiu" "$nomNou"
cp -r "$nomNou" .
rm -r "$nomNou"
else
cp "$arxiu" "$nomNou"
cp "$nomNou" .
rm "$nomNou"
fi
fi
if [ -d "$arxiu" ] && [ $i -ne 2 ]
then cp -r "$arxiu" .
else
if [ $i -ne 2 ]
then cp "$arxiu" .
fi
fi
fi
let comptador=comptador+1
done
tar -cpzf "$comprimit" *
cd ..
rm -r "$temporal"
echo "S'ha comprimit correctament"
exit 0
fi
```

1.3. Joc de proves

Aquesta és la carpeta que hem fet servir per verificar que el nostre script funciona correctament:

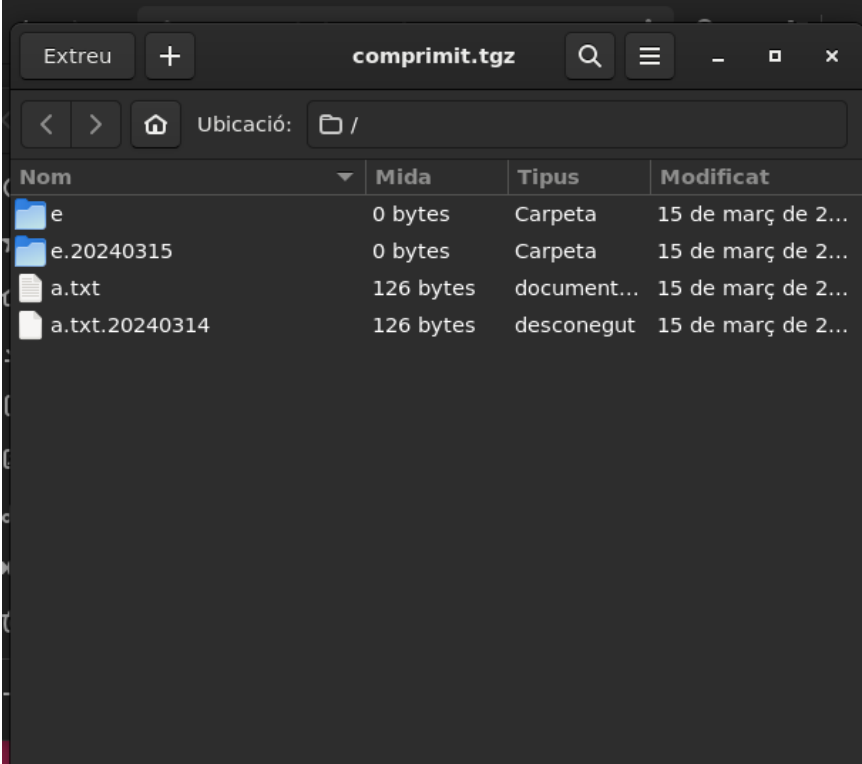


1) Fer còpia de seguretat de dos arxius :



2) Fer còpia de seguretat con filtre “-K” activat:

```
milax@casa: ~/Escriptori/FSO_PRAC1
milax@casa:~/Escriptori/FSO_PRAC1$ ./Copia -K /home/milax/Escriptori/JocdeProves1/a.txt /home/milax/Escriptori/JocdeProves1/e /home/milax/Escriptori/JocdeProves1/comprimit.tgz
S'ha comprimit correctament
milax@casa:~/Escriptori/FSO_PRAC1$
```



| Nom | Mida | Tipus | Modificat |
|----------------|-----------|-------------|--------------------|
| e | 0 bytes | Carpeta | 15 de març de 2... |
| e.20240315 | 0 bytes | Carpeta | 15 de març de 2... |
| a.txt | 126 bytes | document... | 15 de març de 2... |
| a.txt.20240314 | 126 bytes | desconegut | 15 de març de 2... |

3) Utilitzar el filtre “-D” (DryRun) a la copia de seguretat:

```
milax@casa:~/Escriptori/FSO_PRAC1$ ./Copia -D /home/milax/Escriptori/JocdeProves1/c.txt /home/milax/Escriptori/JocdeProves1/e /home/milax/Escriptori/JocdeProves1/comprimit.tgz
fi
cd "$temporal"
for arxiu in $*
do
    if [ $comptador -ge $i ] && [ $comptador -ne $# ]
    then
        if [ $i -eq 2 ]
        then
            rutaInterna=$(find . -name "${basename "$arxiu"}")
            dir=$(dirname "$rutaInterna")
            data=$(stat -c %y "$arxiu" | cut -d' ' -f1 | sed 's/-//g')
            nomNou="$arxiu.$data"
            if [ -d "$arxiu" ]
            then
                cp -r "$arxiu" "$nomNou"
                cp -r "$nomNou" .
                rm -r "$nomNou"
            else
                cp "$arxiu" "$nomNou"
                cp "$nomNou" .
                rm "$nomNou"
            fi
        fi
        if [ -d "$arxiu" ] && [ $i -ne 2 ]
        then cp -r "$arxiu" .
        else
            if [ $i -ne 2 ]
            then cp "$arxiu" .
            fi
        fi
    fi
    let comptador=comptador+1
done
tar -cpzf "$comprimit" *
cd ..
rm -r "$temporal"
echo "S'ha comprimit correctament"
exit 0
fi
milax@casa:~/Escriptori/FSO_PRAC1$
```

4) Si s'indica un número d'arguments incorrecte:

```
milax@casa:~/Escriptori/FSO_PRAC1$ ./Copia /home/milax/Escriptori/JocdeProves1/d
Necessites mes arguments
milax@casa:~/Escriptori/FSO_PRAC1$
```

2. Recuperació d'un directori o fitxer a un altre directori

Recupera [-D] comprimit.tgz fitxer1 fitxer2 ... dirDesti

2.1. Disseny

S'ha optat a utilitzar la mateixa dinàmica que en l'exercici anterior, de manera que s'empra un directori auxiliar temporal per a emmagatzemar momentàriament els arxius que conté el comprimit. Aquesta manera de fer-ho facilita molt la manipulació de fitxers per al seu tractament.

Una vegada ja s'ha descomprimit el recull de fitxers, es decideix recórrer tot el directori temporal i examinar cada fitxer per a saber si és un fitxer que es vol recuperar o no.

En el cas de trobar-ne un que és sol·licitat, es comprova si es tracta d'un directori o no per a utilitzar el mètode de còpia adequat. Una vegada copiat l'arxiu, s'averigua si n'existeixen duplicats per a copiar-los també.

Una vegada s'ha fet tot això, s'elimina el directori temporal creat al principi i es comunica el correcte funcionament de l'aplicació.

En el cas que s'especifiqui la voluntat de tenir una "Dry Run", s'utilitza la comanda tail per a mostrar el codi pertinent.

2.2. Codi de l'script

```
if [ $# -le 2 ]
then
echo "Necessites més arguments!"

exit 1
fi

if [ $1 = "-D" ]
then
tail -33 $0
exit 0
fi

if [[ -e $1 && ( "$1" = *.tgz || "$1" = *.tar.gz )]]
then
comprimit="$1"
desti="{@: -1}"
temporal="dirTemp"

if [ ! -e $desti ]
then mkdir "$desti"
fi

mkdir "$temporal"

tar -xzf "$comprimit" -C "$temporal"
```



```
cd "$temporal"

for arxiu in $*
do
if [ "$arxiu" != "$desti" ] && [ "$arxiu" != "$comprimit" ]
then
if [ -d "$arxiu" ]
then
cp -r "$arxiu" "$desti"
if [ -e "$arxiu".* ]
then cp -r "$arxiu".* "$desti"
fi
else
cp "$arxiu" "$desti"
if [ -e "$arxiu".* ]
then cp "$arxiu".* "$desti"
fi
fi
done
cd ..

rm -r "$temporal"

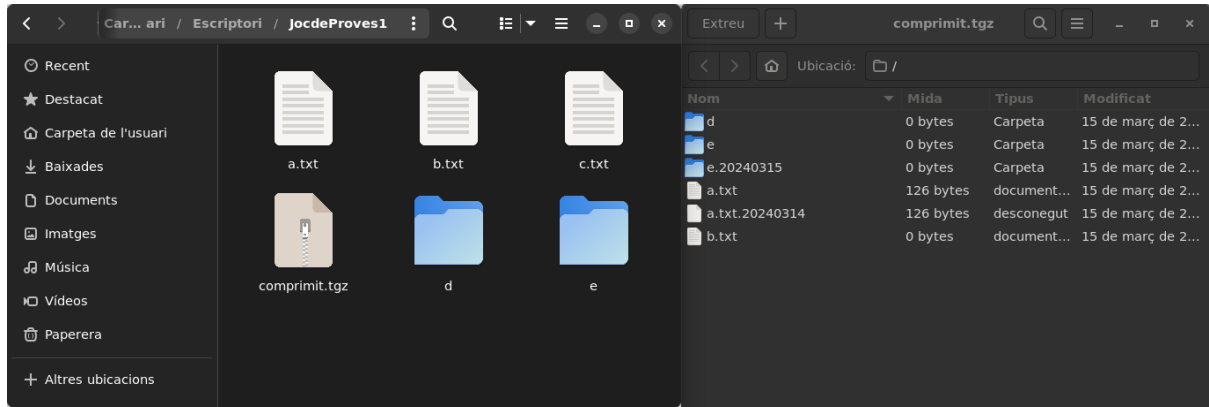
echo "La recuperació ha sigut satisfactoria"

exit 0

fi
```

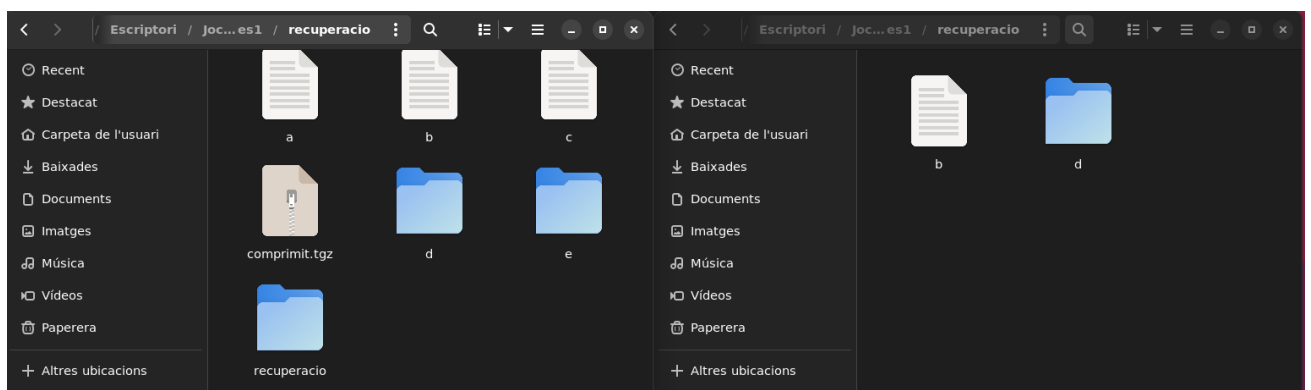
2.3. Joc de proves

Al joc de proves d'aquest script utilitzarem la mateixa carpeta que en l'anterior script , afegint-li el comprimit que volem recuperar, i d'aquesta manera verificar que la nostra recuperació funciona correctament:

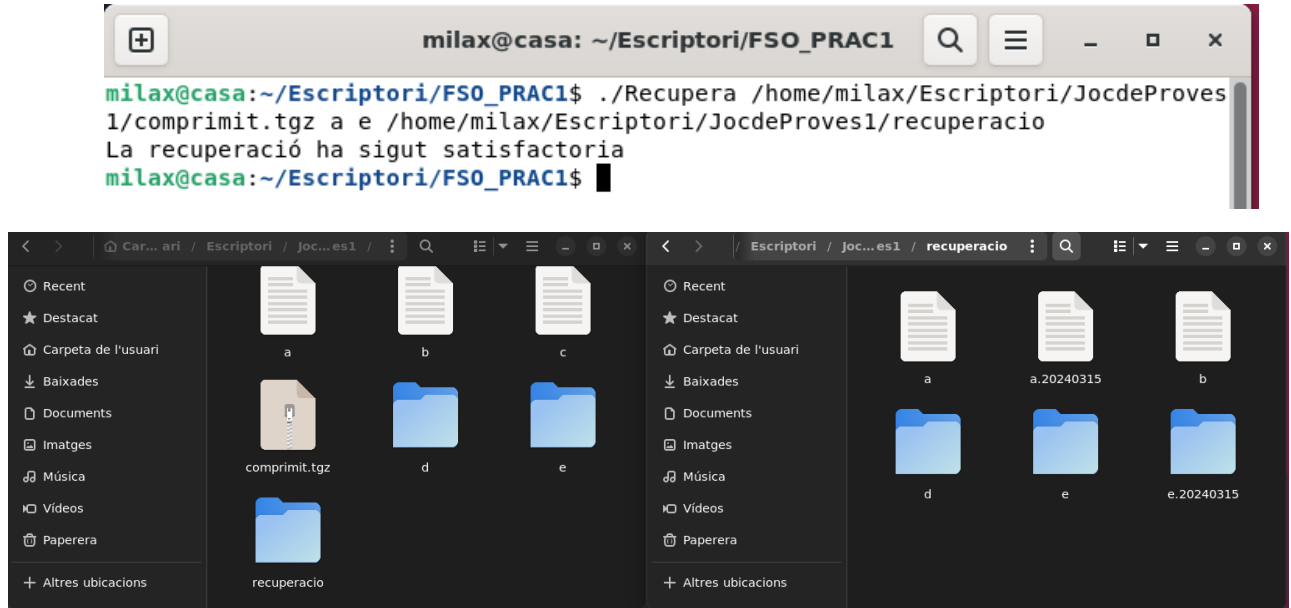


- 1) Comprovarem que la recuperació d'un arxiu no duplicat funciona:

```
milax@casa: ~/Escriptori/FSO_PRAC1
milax@casa:~/Escriptori/FSO_PRAC1$ ./Recupera /home/milax/Escriptori/JocdeProves1/comprimit.tgz b d /home/milax/Escriptori/JocdeProves1/recuperacio
La recuperació ha sigut satisfactoria
milax@casa:~/Escriptori/FSO_PRAC1$
```



- 2) Comprovar que l'script també pot recuperar arxius que estiguin duplicats:



- 3) Comrpovar que l'script funciona indicant el filtre "-D" (DryRun):

```
milax@casa:~/Escriptori/FSO_PRAC1$ ./Recupera -D /home/milax/Escriptori/JocdeProves1/comprimit.tgz a e /home/milax/Escriptori/JocdeProves1/recuperacio
then
  comprimit="$1"
  desti="{@: -1}"
  temporal="dirTemp"
  if [ ! -e $desti ]
  then mkdir "$desti"
  fi
  mkdir "$temporal"
  tar -xzf "$comprimit" -C "$temporal"
  cd "$temporal"
  for arxiu in $*
  do
    if [ "$arxiu" != "$desti" ] && [ "$arxiu" != "$comprimit" ]
    then
      if [ -d "$arxiu" ]
      then
        cp -r "$arxiu" "$desti"
        if [ -e "$arxiu".* ]
        then cp -r "$arxiu".* "$desti"
        fi
      else
        cp "$arxiu" "$desti"
        if [ -e "$arxiu".* ]
        then cp "$arxiu".* "$desti"
        fi
      fi
    fi
  done
  cd ..
  rm -r "$temporal"
  echo "La recuperació ha sigut satisfactoria"
  exit 0
fi
milax@casa:~/Escriptori/FSO_PRAC1$
```

- 4) Verificar que l'usuari indica el número d'arguments correcte:

```
milax@casa:~/Escriptori/FSO_PRAC1$ ./Recupera /home/milax/Escriptori/JocdeProves1/comprimit.tgz a
Necessites més arguments!
```

3. Automatització amb crontab

Configura dia hora script fitxer1 ... fitxer_tgz

3.1. Disseny

Aquest exercici no dona molta oportunitat a innovacions en quant a disseny. S'ha optat a fer la comprovació de que es reben arguments vàlids: fitxers existents o format de dia i hora adequat.

Una vegada se sap que l'usuari ha cridat l'script de forma adequada, s'utilitza la comanda crontab per a establir una còpia de seguretat amb la freqüència determinada per l'usuari.

echo "\$minuts \$hora \$dia * * \$nomScript \$resta" | crontab -

El guió que segueix a la comanda crontab indica que la informació prové d'una línia de codi anterior.

3.2. Codi de l'script

```
if [ "$#" -lt 5 ]
then
    echo "Necessites mes arguments"
    exit 1
fi

dia="$1"
hora=$(echo "$2" | cut -d':' -f1)
minuts=$(echo "$2" | cut -d':' -f2)
nomScript="$3"
if [ ! -x "$nomScript" ]
then
    echo "No existeix l'script especificat."
    exit 2
fi

for argument in "${@:4}"
do resta+="$argument"
done

echo "$minuts $hora $dia * * $nomScript $resta" | crontab -
echo "Copia de seguretat establerta correctament a $dia, $hora:$minuts."
```

3.3. Joc de proves

```
milax@casa: ~/Escriptori/FSO_PRAC1
milax@casa:~/Escriptori/FSO_PRAC1$ ./Configura 15 17:34 /home/milax/Escriptori/FSO_PRAC1/Copia
/home/milax/Escriptori/JocdeProves1/b /home/milax/Escriptori/prueba.tgz
Copia de seguretat establerta correctament a 15, 17:34.
milax@casa:~/Escriptori/FSO_PRAC1$
```

```
GNU nano 7.2 /tmp/crontab.jz7dxe/crontab
34 17 15 * * /home/milax/Escriptori/FSO_PRAC1/Copia 1517:34/home/milax/Escriptori/FSO_PRAC1/C>
```