Javier Pedragosa

ACTIVITAT 6 – BASH and Crontab

1. BASH:

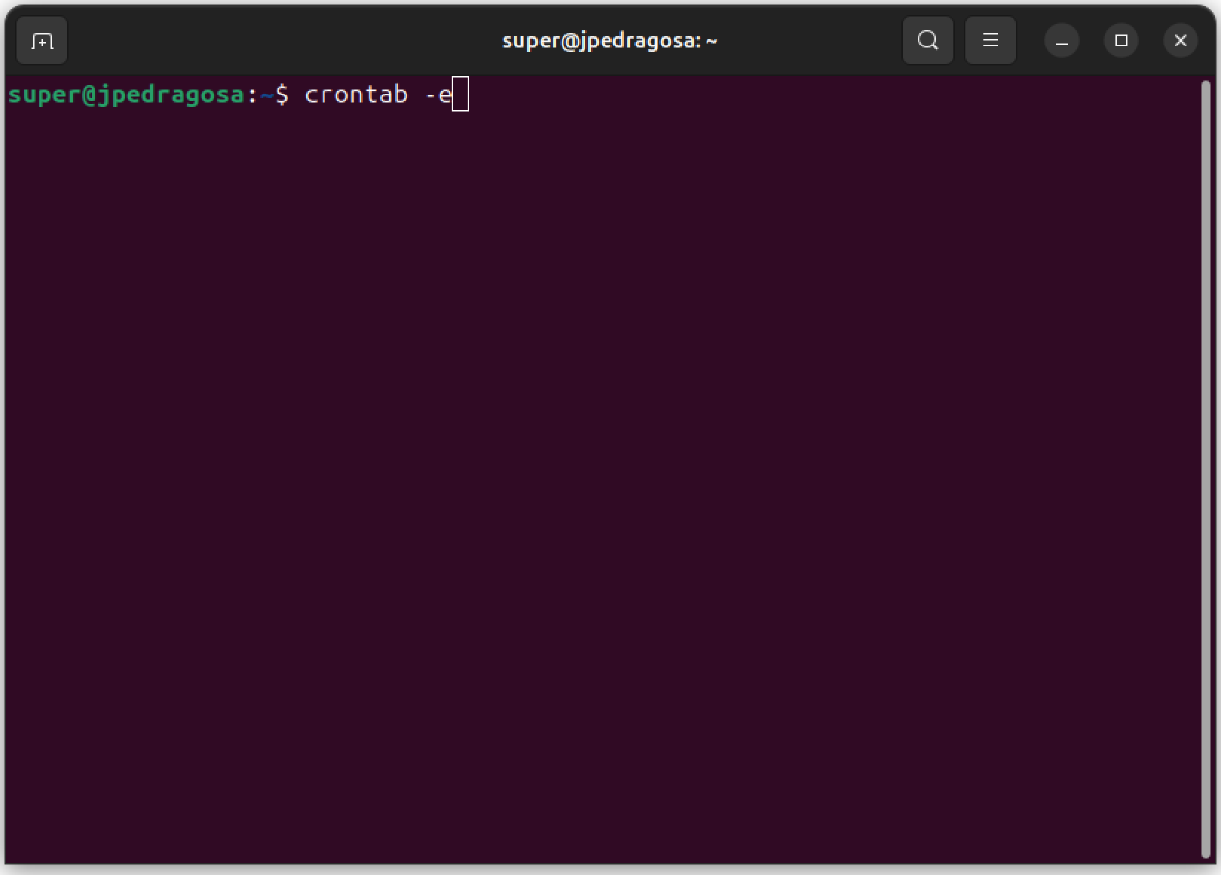
```
#!/bin/bash
#Constants
re='^[0-9]+$'
possibleHands=("pedra" "paper" "tisora")

#Declaració variables
name=""
running=1
ronda=1
targetPoints=0
playerPoints=0
cpuPoints=0
playerHand=""
cpuHand=""
missatgeGuanyador=""

#Funcions
#Aquesta funció defineix aleatoriament la mà de l'ordinador
ordinador_tria() {
cpuHand=${possibleHands[$(( RANDOM % 3 ))]}
}

#Aquesta funció compara la mà del jugador amb la de l'ordinador, envia el missatge pertinent i
suma punts a qui correspongui
determinar_tirada() {
case $cpuHand in
pedra)
case $playerHand in
pedra)
echo "L'ordinador ha jugat $cpuHand. Empat!"
;;
paper)
echo "L'ordinador ha jugat $cpuHand. Guanyes la ronda!"
((playerPoints++))
;;
tisora)
echo "L'ordinador ha jugat $cpuHand. Perds la ronda!"
((cpuPoints++))
;;
x)
running=0
;;
```

```
esac
;;
paper)
case $playerHand in
paper)
echo "L'ordinador ha jugat $cpuHand. Empat!"
;;
tisora)
echo "L'ordinador ha jugat $cpuHand. Guanyes la ronda!"
((playerPoints++))
;;
pedra)
echo "L'ordinador ha jugat $cpuHand. Perds la ronda!"
((cpuPoints++))
;;
x)
running=0
;;
esac
;;
tisora)
case $playerHand in
tisora)
echo "L'ordinador ha jugat $cpuHand. Empat!"
;;
pedra)
echo "L'ordinador ha jugat $cpuHand. Guanyes la ronda!"
((playerPoints++))
;;
paper)
echo "L'ordinador ha jugat $cpuHand. Perds la ronda!"
((cpuPoints++))
;;
x)
running=0
;;
esac
;;
esac
}
```

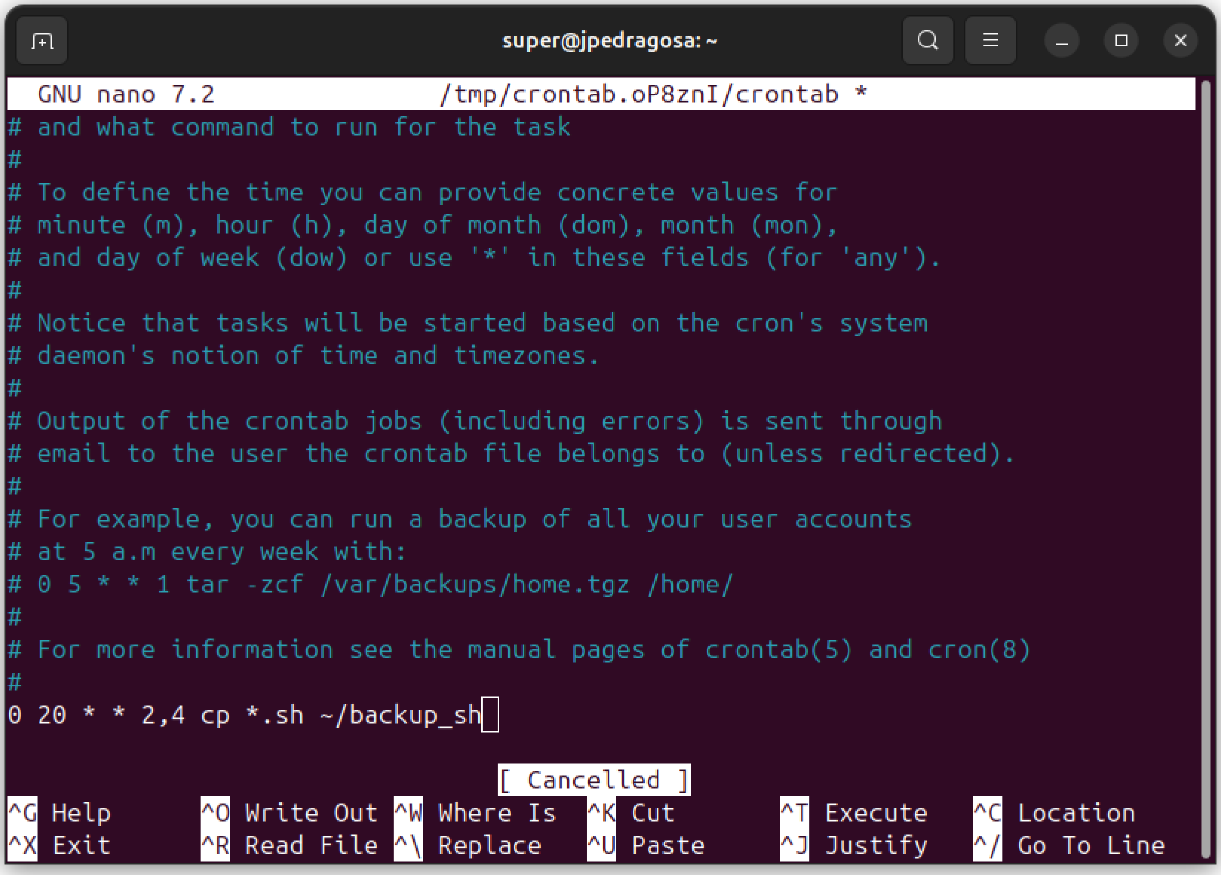
```
#Aquesta funció comprova si algú ha guanyat comparant la quantitat de punts amb els punts objectiu
algu_guanyat () {
if [ "$playerPoints" == "$targetPoints" ]
then
running=0
missatgeGuanyador="Guanyes tú, $name!"
elif [ "$cpuPoints" == "$targetPoints" ]
then
running=0
missatgeGuanyador="Guanya l'ordinador..."
elif [ "$cpuPoints" == "$playerPoints" ]
then
missatgeGuanyador="Empat!"
else
if [ "$playerPoints" -gt "$cpuPoints" ]
then
missatgeGuanyador="Guanyes tú, $name!"
else
missatgeGuanyador="Guanya l'ordinador..."
fi
fi
}


#Inici, on demanem el nom del jugador i la quantitat de rondes a jugar.
clear
echo "Pedra, paper o tisores, programat per Javier Pedragosa."
while [ "$name" == "" ] | | [[ $name =~ $re ]]
do
echo "Si us plau, introdueix el teu nom: "
read -r name
clear
done

echo "Benvingut/da, $name."

while ! [[ $targetPoints =~ $re ]] | | [ "$targetPoints" -lt 1 ]
do
echo "Quants punts vols jugar?"
read -r targetPoints
clear
done

#Imprimim les instruccions del joc.
echo "Per jugar pedra, introdueix 'pedra'."
echo "Per jugar paper, introdueix 'paper'."
echo "Per jugar tisora, introdueix 'tisora'."
echo "Per finalitzar el joc abans de temps, introdueix 'x'."
echo ""
echo "Iniciar joc (intro)?"
read -r
clear
#Bucle principal del joc.
while [ "$running" == 1 ]
do
echo "Ronda $ronda. $name $playerPoints - $cpuPoints CPU"
while [ "$playerHand" != "pedra" ] && [ "$playerHand" != "paper" ] && [ "$playerHand" != "tisora" ]
&& [ "$playerHand" != "x" ]
```

```
do
echo "Quina mà vols jugar, $name? ('pedra', 'paper' o 'tisora' (o 'x' per sortir))"
read -r playerHand
done
ordinador_tria
determinar_tirada
playerHand=""
echo "[INTRO per continuar]"
read -r
clear
algu_guanyat
((ronda++))
done

echo "Resultat després de $ronda rondes:"
echo "$name $playerPoints - $cpuPoints CPU"
echo "$missatgeGuanyador"
echo "INTRO per sortir"
read -r
```

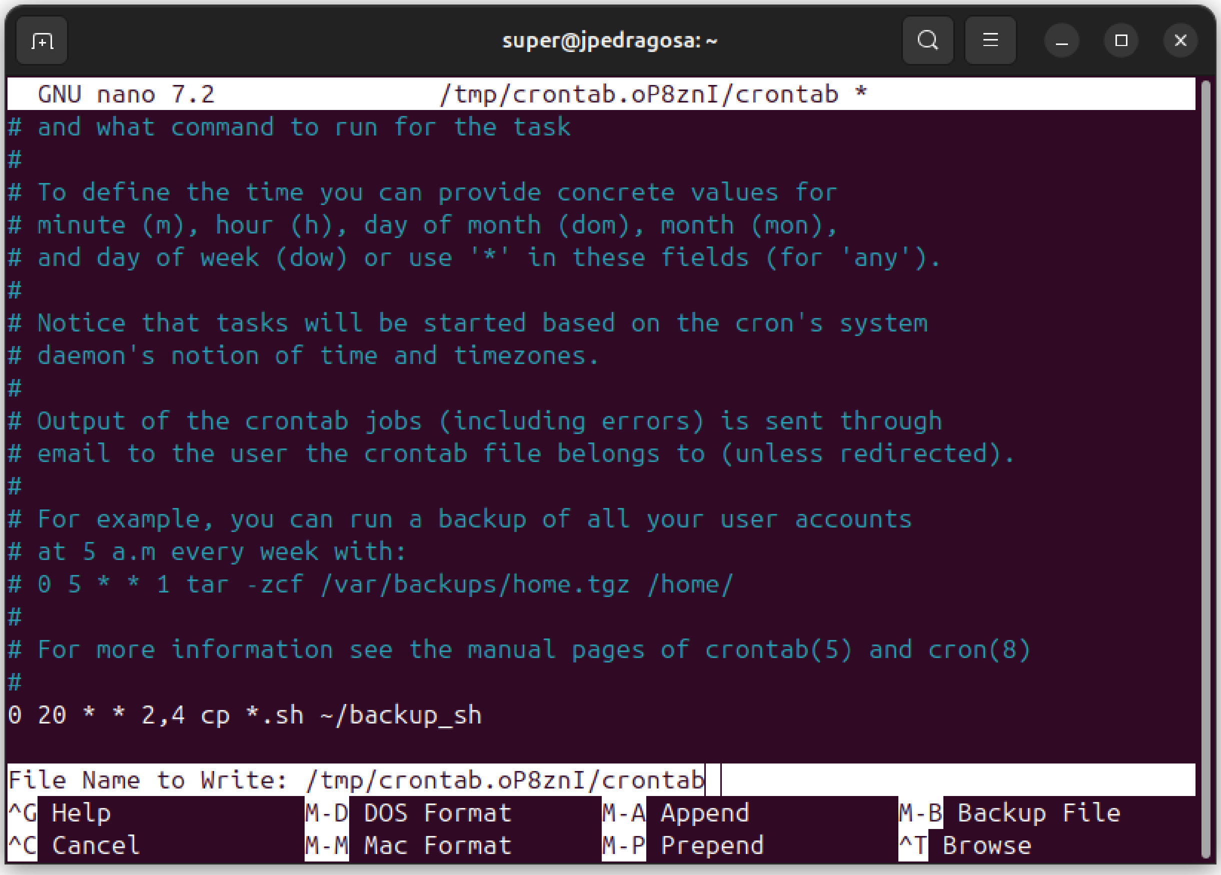
2. CRONTAB:



```
super@jpedragosa:~$ crontab -e
```



```
  GNU nano 7.2                /tmp/crontab.oP8znI/crontab *
# and what command to run for the task
#
# To define the time you can provide concrete values for
# minute (m), hour (h), day of month (dom), month (mon),
# and day of week (dow) or use '*' in these fields (for 'any').
#
# Notice that tasks will be started based on the cron's system
# daemon's notion of time and timezones.
#
# Output of the crontab jobs (including errors) is sent through
# email to the user the crontab file belongs to (unless redirected).
#
# For example, you can run a backup of all your user accounts
# at 5 a.m every week with:
# 0 5 * * 1 tar -zcf /var/backups/home.tgz /home/
#
# For more information see the manual pages of crontab(5) and cron(8)
#
0 20 * * 2,4 cp *.sh ~/backup_sh

                                             [ Cancelled ]
^G Help         ^O Write Out ^W Where Is  ^K Cut        ^T Execute   ^C Location
^X Exit         ^R Read File ^\ Replace   ^U Paste      ^J Justify   ^/ Go To Line
```



```
  GNU nano 7.2               /tmp/crontab.oP8znI/crontab *
# and what command to run for the task
#
# To define the time you can provide concrete values for
# minute (m), hour (h), day of month (dom), month (mon),
# and day of week (dow) or use '*' in these fields (for 'any').
#
# Notice that tasks will be started based on the cron's system
# daemon's notion of time and timezones.
#
# Output of the crontab jobs (including errors) is sent through
# email to the user the crontab file belongs to (unless redirected).
#
# For example, you can run a backup of all your user accounts
# at 5 a.m every week with:
# 0 5 * * 1 tar -zcf /var/backups/home.tgz /home/
#
# For more information see the manual pages of crontab(5) and cron(8)
#
0 20 * * 2,4 cp *.sh ~/backup_sh

File Name to Write: /tmp/crontab.oP8znI/crontab
^G Help              M-D DOS Format       M-A Append           M-B Backup File
^C Cancel            M-M Mac Format       M-P Prepend          ^T Browse
```