

ADSO Training 6 Temporització de tasques

Índex

1. Objectiu	2
2. Abans de començar	2
3. Tasques puntuals	2
4. Tasques periòdiques	3

1. Objectiu

Ser capaçes de programar tasques de manera puntual i periòdica.

2. Abans de començar

Hauríeu de tenir fets els següents scripts de les pràctiques anteriors:

ocupacio.sh

badusers.py

Comprovem si tenim els scripts:

```
root@Victor (Thu Dec 05) >:~/Downloads#ls
BadUsers.py          class_act.py          jdk-6u45-linux-x64.bin
BadUsers.sh          class_act.sh          ocupacio.py
Spyder-Linux-x86_64.sh infouser.py           ocupacio.sh
asosh.py             infouser.sh          user-stats.py
borrar_users.sh      jdk-1_5_0_22-linux-amd64.bin user-stats.sh
root@Victor (Thu Dec 05) >:~/Downloads#
```

3. Tasques puntuals

Amb quina comanda es programen habitualment les tasques puntuals?

Les tasques puntuals es programen amb la comanda *at*. Per usar-la, hem de descarregar-la.

apt update

```
root@Victor (Fri Dec 06) >:~#sudo apt update
Hit:1 http://deb.debian.org/debian stable InRelease
Hit:2 http://ftp.es.debian.org/debian stable InRelease
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
64 packages can be upgraded. Run 'apt list --upgradable' to see them.
N: Repository 'Debian bookworm' changed its 'non-free component' value from 'non-free' to 'non-free non-free-firmware'
N: More information about this can be found online in the Release notes at: http://www.debian.org/releases/bookworm/amd64/release-notes/ch-information.html#non-free-split
root@Victor (Fri Dec 06) >:~#
```

apt install at

```
root@Victor (Fri Dec 06) >:~#apt install at
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following packages were automatically installed and are no longer required:
bsdmainutils libaom0 libbpf0 libcupsfilters1 libcupsimage2 libdav1d4 libdns-export1110 libflac8 libicu63 libicu67 liblmbase23 liblmbase25 libisc-export1105 libldap-2.4-2
libopenexr23 libopenexr25 libperl5.28 libperl5.32 libprocps8 libpython3.7-minimal libpython3.7-stdlib libpython3.9-minimal libpython3.9-stdlib libwmf-0.2-7 libwmf0.2-7 lib
linux-image-4.19.0-21-amd64 linux-image-4.19.0-6-amd64 perl-modules-5.28 perl-modules-5.32 pigz python3.7-minimal python3.9 python3.9-minimal
Use 'apt autoremove' to remove them.
The following additional packages will be installed:
bsd-mailx exim4-base exim4-config exim4-daemon-light libgnutls-dane0 libidn12 liblockfile-bin liblockfile1 libunbound8 psmisc
Suggested packages:
exim4-doc-html | exim4-doc-info eximon4 spf-tools-perl swaks dns-root-data
The following NEW packages will be installed:
at bsd-mailx exim4-base exim4-config exim4-daemon-light libgnutls-dane0 libidn12 liblockfile-bin liblockfile1 libunbound8 psmisc
0 upgraded, 11 newly installed, 0 to remove and 64 not upgraded.
Need to get 3450 kB of archives.
After this operation, 7584 kB of additional disk space will be used.
Do you want to continue? [Y/n] y
```

Un cop instal·lat, hem d'activar el servei atd.

sudo systemctl enable atd

```
root@Victor (Fri Dec 06) >:~#systemctl enable atd
Synchronizing state of atd.service with SysV service script with /lib/systemd/systemd-sysv-install.
Executing: /lib/systemd/systemd-sysv-install enable atd
perl: warning: Setting locale failed.
perl: warning: Please check that your locale settings:
    LANGUAGE = (unset),
    LC_ALL = (unset),
    LANG = "es_ES.UTF-8"
    are supported and installed on your system.
perl: warning: Falling back to the standard locale ("C").
perl: warning: Setting locale failed.
perl: warning: Please check that your locale settings:
    LANGUAGE = (unset),
    LC_ALL = (unset),
    LANG = "es_ES.UTF-8"
    are supported and installed on your system.
perl: warning: Falling back to the standard locale ("C").
root@Victor (Fri Dec 06) >:~#systemctl enable atd
```

sudo systemctl start atd

```
root@Victor (Fri Dec 06) >:~#systemctl start atd
root@Victor (Fri Dec 06) >:~#
```

Verifiquem l'estat

sudo systemctl status atd

```
root@Victor (Fri Dec 06) >:~#systemctl status atd
â atd.service - Deferred execution scheduler
   Loaded: loaded (/lib/systemd/system/atd.service; enabled; preset: enabled)
   Active: active (running) since Fri 2024-12-06 13:15:47 UTC; 2min 40s ago
     Docs: man:atd(8)
    Main PID: 1334 (atd)
      Tasks: 1 (limit: 2294)
     Memory: 252.0K
        CPU: 2ms
    CGroup: /system.slice/atd.service
            â1334 /usr/sbin/atd -f

Dec 06 13:15:47 aso-client systemd[1]: Starting atd.service - Deferred execution scheduler...
Dec 06 13:15:47 aso-client systemd[1]: Started atd.service - Deferred execution scheduler.
root@Victor (Fri Dec 06) >:~#
```

Fent servir aquesta comanda programeu les següents tasques:

Un esborrat dels fitxers del directori /tmp a les 2 de la nit d'avui.

Per programar aquesta comanda executem a la terminal

echo "rm -rf /tmp/*" | at 02:00

```
root@Victor (Fri Dec 06) >:~#echo "rm -rf /tmp/*" | at 02:00
warning: commands will be executed using /bin/sh
job 1 at Sat Dec 7 02:00:00 2024
root@Victor (Fri Dec 06) >:~#
```

VOISEA

Si volem consultar les tasques que hem programat les podem veure amb la comanda `atq`.

```
root@Victor (Fri Dec 06) >:"#atq
1      Sat Dec  7 02:00:00 2024 a root
root@Victor (Fri Dec 06) >:"#
```

Fer un llistat dels usuaris connectats al servidor dintre de 10 minuts. Aquest llistat s'ha de deixar al home del root amb el nom `usuaris-data.txt`, on data serà la data del moment en que s'ha fet el llistat.

Executem la següent comanda:

`echo "who > /root/usuaris-$(date '+%Y-%m-%d_%H-%M').txt" | at now + 10 minutes`

```
root@Victor (Fri Dec 06) >:"#echo "who > /root/usuaris-$(date '+%Y-%m-%d_%H-%M').txt" | at now + 10 minutes
warning: commands will be executed using /bin/sh
job 2 at Fri Dec  6 13:41:00 2024
root@Victor (Fri Dec 06) >:"#atq
2      Fri Dec  6 13:41:00 2024 a root
1      Sat Dec  7 02:00:00 2024 a root
root@Victor (Fri Dec 06) >:"#
```

Quan hagin passat els deu minuts, podem comprovar que s'ha creat el fitxer i que conté els usuaris actius.

Un shutdown del vostre servidor al final de la classe. Volem que avisi als usuaris del sistema 5 minuts abans.

La comanda que utilitzarem és

`echo "shutdown -h now" | at 10:30`

Suposant que la classe acaba a les 10.30

I per fer l'avís podem crear una altra tasca amb la comanda:

`echo "wall 'El sistema s'apagarà en 5 minuts.'" | at 10:25`

```
root@Victor (Fri Dec 06) >:"#echo "shutdown -h now" | at 10.30
warning: commands will be executed using /bin/sh
job 3 at Sat Dec  7 10:30:00 2024
root@Victor (Fri Dec 06) >:"#echo "wall 'El sistema s'apagar en 5 minuts.'" | at 10.25
warning: commands will be executed using /bin/sh
job 4 at Sat Dec  7 10:25:00 2024
root@Victor (Fri Dec 06) >:"#
```

O també ho podem fer en una sola comanda

`echo "wall 'El sistema s'apagarà en 5 minuts. '; sleep 300; shutdown -h now" | at 15:25`

On el que fem es executar el missatge a l'hora seleccionada amb la comanda at, després un sleep de 300 segons (5 minuts), i l'execució del shutdown.

```
job 4 at Sat Dec 7 10:25:00 2024
root@Victor (Fri Dec 06) >:"#echo "wall 'El sistema s'apagara en 5 minuts.'; sleep 300; shutdown -h now" | at 10,15
warning: commands will be executed using /bin/sh
job 5 at Sat Dec 7 10:15:00 2024
root@Victor (Fri Dec 06) >:"#
```

Volem restringir aquest servei a l'usuari root.

Quin fitxer hem de crear? Amb quin contingut?

Per restringir el servei als usuaris, podem modificar dos fitxers.

El primer és el fitxer at.allow, que llista als usuaris que poden utilitzar aquest servei.

El segon és el fitxer at.deny, que llista als usuaris que no tenen el permís per utilitzar aquest servei.

"At" mira primer el fitxer at.allow, i si existeix no mira el fitxer at.deny, per tant només hem de crear el fitxer at.allow i posar el nom de root.

```
GNU nano 7.2 /etc/at.allow
root
```

Proveu de fer servir la comanda at amb un altre usuari i comproveu que efectivament no us deixa.

Provem que funcioni amb root:

```
root@Victor (Fri Dec 06) >:"#echo "echo 'Prova correcte' > /tmp/prova.txt" | at now
warning: commands will be executed using /bin/sh
job 7 at Fri Dec 6 13:54:00 2024
root@Victor (Fri Dec 06) >:"#
```

```
job 7 at Fri Dec 6 13:54:00 2024
root@Victor (Fri Dec 06) >:"#cd /tmp/
root@Victor (Fri Dec 06) >:"#ls
prova.txt      systemd-private-9974585a58ee43198f67edaa107317c8-colord.service-djj66T
ssh-XXXXXX1GPlj systemd-private-9974585a58ee43198f67edaa107317c8-systemd-logind.service-3vFKX1
root@Victor (Fri Dec 06) >:"#
```

```
GNU nano 7.2 prova.txt
Prova correcte
```

Amb root hem comprovat que funciona correcte, ara fem login amb altre usuari i provem.

```
user1@aso-client:~$ echo "echo 'Prova incorrecte' > /tmp/prova.txt" | at now
You do not have permission to use at.
user1@aso-client:~$
```

I podem comprovar que amb user1 no tenim permisos per utilitzar el servei at.

4. Tasques periòdiques

Mireu el manual de les comandes cron i crontab i contesteu a les següents preguntes:

Com podem veure quin es el contingut del cron d'un usuari?

crontab -l

Com podem afegir-hi noves tasques periòdiques?

crontab -e

```
SergiS@aso-client:~$ crontab -e
no crontab for SergiS - using an empty one

Select an editor. To change later, run 'select-editor'.
 1. /bin/nano      <---- easiest
 2. /usr/bin/vim.basic
 3. /usr/bin/vim.tiny

Choose 1-3 [1]:
```

```
GNU nano 7.2 /tmp/crontab.Z7504L/crontab
# Edit this file to introduce tasks to be run by cron.
#
# Each task to run has to be defined through a single line
# indicating with different fields when the task will be run
# 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/
#
[ Read 25 lines ]
^G Help      ^O Write Out ^W Where Is  ^K Cut       ^T Execute   ^C Location
^X Exit      ^R Read File ^N Replace   ^U Paste     ^J Justify   ^_ Go To Line
```

Com podem limitar qui té accés al servei de cron?

modifiquen els fitxers cron.allow i cron.deny

Un exemple a la hora de fer us d'aquestes comandes esmentades pot ser per permetre l'ús únicament a l'usuari root.

```
root@oscarMP (Tue Dec 10):~$ # echo "root" | sudo tee /etc/cron.allow
root
```

Un problema a l'hora de fer servir cron és assegurar-nos que hem especificat correctament la freqüència sense haver d'esperar i veure que quan arriba el moment no s'executa (en particular quan falta molt de temps).

Que podríem fer per comprovar en el mateix moment que els nostres fitxers de crontab són correctes?

Podem utilitzar la comanda **crontab -l** per revisar el contingut del crontab abans de guardar-lo. A més, podem validar la sintaxi de les tasques cron amb eines en línia o scripts que verifiquin la estructura dels camps de temps.

o fem que la comanda s'executi en menys temps del que ha de ser per a comprobarlo, en comptes de esperar un any, posem que s'executi d'aquí a 10 minuts i així ho comprovem

Quins efectes col·laterals tindria això?

La validació immediata de les tasques cron pot prevenir errors de programació que podrien causar que les tasques no s'executïn correctament. No obstant això, una validació automàtica o l'ús d'eines de verificació podria introduir una complexitat addicional o requerir permisos especials, i si la validació falla, podria impedir la programació de tasques legítimes.

Volem controlar l'ocupació d'una sèrie d'usuaris (fent servir l'script ocupacio.sh que vareu fer) que estaran especificats al fitxer /etc/ocupacio.users. El control es farà tots els diumenges.

Quina entrada afegireu al crontab? De quin usuari?

afegiríem: 0 0 * * 0 /Downloads/ocupacio.sh al crontab de l'usuari root

```
root@SergiC(Sun Dec 08):~# crontab -e
```

```
GNU nano 7.2 /tmp/crontab.RTDLp0/crontab
#
# 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)
#
# m h dom mon dow  command
0 0 * * 0 /Downloads/ocupacio.sh
```


VOISEA

```
root@SergiC(Sun Dec 08):~# crontab -e
no crontab for root - using an empty one
crontab: installing new crontab
root@SergiC(Sun Dec 08):~#
```

També volem controlar els usuaris innecessaris del sistema fent servir l'script badusers.py. Aquest control es realitzarà el primer dia de cada mes.

Quina entrada afegireu al crontab? De quin usuari?

```
root@SergiC(Sun Dec 08):/etc# crontab -e
```

```
GNU nano 7.2 /tmp/crontab.qpZzIJ/crontab
# 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)
#
# m h dom mon dow command
0 0 * * 0 Downloads/ocupacio.sh
0 0 1 * * Downloads/BadUsers.sh
```

```
root@SergiC(Sun Dec 08):/etc# crontab -e
crontab: installing new crontab
```

Que hem de fer perquè el llistat d'usuaris que genera l'script ens arribi per correu?

```
GNU nano 7.2 /tmp/crontab.qM6t50/crontab *
# 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)
#
# m h dom mon dow command
0 2 * * 0 Downloads/ocupacio.sh
0 3 1 * * /usr/bin/python3 Downloads/badUsers.py | mail -s "Llistat de BadUsers" oscar.marquez.puchol@estudiantat.upc.edu
```

L'usuari sergiS vol fer backups del seu home. Feu servir cron perquè es compleixin les següents condicions:

És farà un backup total cada primer dilluns de mes

És farà un backup incremental tots els dimecres i dissabtes respecte al backup total.

Els backups es guardaran al mateix home de l'usuari sergiS a un directori backups. Evidentment no s'ha de fer backup dels backups.

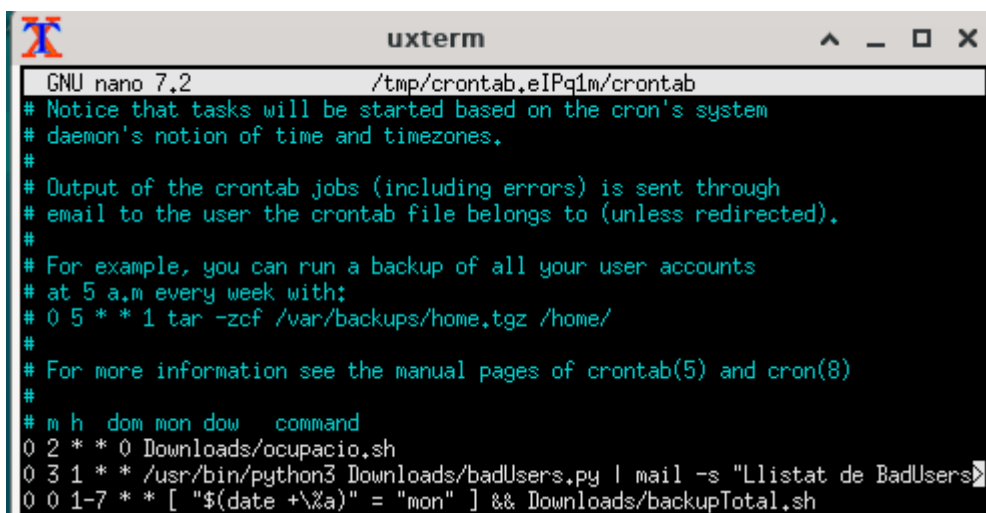
El nom dels backups ha d'incloure la data en la que es varen fer i el tipus de backup.

Per reduir l'espai que ocupen el primer dia de cada mes es comprovarà si la mida total dels backups és superior a 100 Mb. Si és així s'esborraran tants backups com calgui fins estar per sota del límit en ordre cronològic ascendent (primer els més vells).

L'usuari sergiS només vol rebre missatges de cron si hi ha errors.

Mostreu com ho heu fet.

És farà un backup total cada primer dilluns de mes

A screenshot of a terminal window titled 'uxterm'. Inside, the GNU nano 7.2 editor is open, editing a file at /tmp/crontab.eIPqIm/crontab. The file contains a crontab configuration with comments and three scheduled tasks. The tasks are: a weekly backup at 5 AM, a daily task at 3 PM, and a monthly backup on the 1st of each month. The monthly task is highlighted with a mouse cursor.

```
GNU nano 7.2 /tmp/crontab.eIPqIm/crontab
# 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)
#
# m h dom mon dow   command
0 2 * * 0 Downloads/ocupacio.sh
0 3 1 * * /usr/bin/python3 Downloads/badUsers.py | mail -s "Llistat de BadUsers"
0 0 1-7 * * [ "$(date +%a)" = "mon" ] && Downloads/backupTotal.sh
```

C/C++

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

```
USER_HOME="/home/sergiS"
```

```
BACKUP_DIR="$USER_HOME/backups"
```

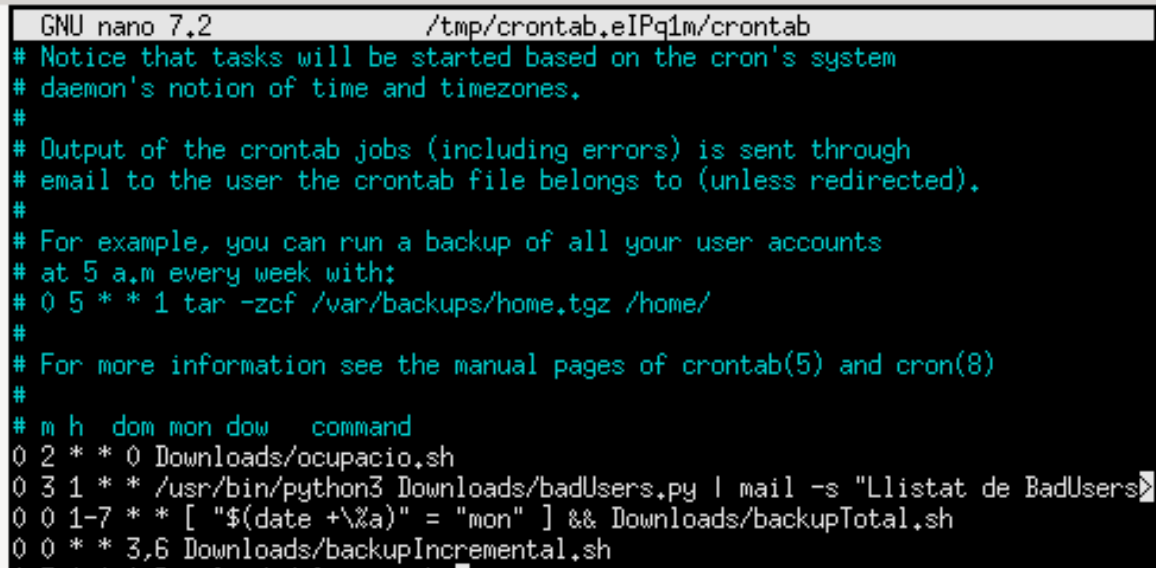
```
DATE=$(date '+%Y-%m-%d')
```

VOISEA

```
BACKUP_NAME="backup_total_$(date +%Y-%m-%d).tar.gz"

mkdir -p "$BACKUP_DIR"
tar --exclude="$BACKUP_DIR" -czf "$BACKUP_DIR/$BACKUP_NAME" "$USER_HOME"
echo "Backup complet realitzat: $BACKUP_NAME"
```

Es farà un backup incremental tots els dimecres i dissabtes respecte al backup total.



```
GNU nano 7.2 /tmp/crontab.eIPq1m/crontab
# 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)
#
# m h dom mon dow   command
0 2 * * 0 Downloads/ocupacio.sh
0 3 1 * * /usr/bin/python3 Downloads/badUsers.py | mail -s "Llistat de BadUsers"
0 0 1-7 * * [ "$(date +%a)" = "mon" ] && Downloads/backupTotal.sh
0 0 * * 3,6 Downloads/backupIncremental.sh
```

C/C++

```
#!/bin/bash

# Variables
USER_HOME="/home/sergiS"
BACKUP_DIR="$USER_HOME/backups"
DATE=$(date +%Y-%m-%d)
BACKUP_NAME="backup_incremental_$(date +%Y-%m-%d).tar.gz"
```

```

    SNAPSHOT_FILE="$BACKUP_DIR/snapshot.file"

    mkdir -p "$BACKUP_DIR"

    tar --exclude="$BACKUP_DIR" --listed-incremental="$SNAPSHOT_FILE" -czf
"$BACKUP_DIR/$BACKUP_NAME" "$USER_HOME"

    echo "Backup incremental realitzat: $BACKUP_NAME"

```

Per reduir l'espai que ocupen el primer dia de cada mes es comprovarà si la mida total dels backups és superior a 100 Mb. Si és així s'esborraran tants backups com calgui fins estar per sota del límit en ordre cronològic ascendent (primer els més vells).

```

GNU nano 7.2 /tmp/crontab.eIPq1m/crontab
# 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)
#
# m h dom mon dow   command
0 2 * * 0 Downloads/ocupacio.sh
0 3 1 * * /usr/bin/python3 Downloads/badUsers.py | mail -s "Llistat de BadUsers"
0 0 1-7 * * [ "$(date +%a)" = "mon" ] && Downloads/backupTotal.sh
0 0 * * 3,6 Downloads/backupIncremental.sh
0 3 1 * * Downloads/cleanup.sh

```

C/C++

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

VOISEA

```
USER_HOME="/home/sergiS"
BACKUP_DIR="$USER_HOME/backups"
MAX_SIZE_MB=100

CURRENT_SIZE_MB=$(du -sm "$BACKUP_DIR" | cut -f1)

if [ "$CURRENT_SIZE_MB" -gt "$MAX_SIZE_MB" ]; then
    echo "Espai actual: ${CURRENT_SIZE_MB}MB. Eliminant backups antics fins
reduir a ${MAX_SIZE_MB}MB..."

    for BACKUP_FILE in $(ls -t "$BACKUP_DIR"); do
        rm "$BACKUP_DIR/$BACKUP_FILE"
        CURRENT_SIZE_MB=$(du -sm "$BACKUP_DIR" | cut -f1)

        if [ "$CURRENT_SIZE_MB" -le "$MAX_SIZE_MB" ]; then
            break
        fi
    done

    echo "Neteges completes. Espai actual: $(du -sm "$BACKUP_DIR" | cut
-f1)MB."
else
    echo "Espai actual: ${CURRENT_SIZE_MB}MB. No cal eliminar res."
fi
```