

**Índex**

[1. Objectiu 2](#_gjdgxs)

[2. Abans de començar 2](#_30j0zll)

[3. Tasques puntuals 2](#_1fob9te)

[4. Tasques periòdiques 3](#_3znysh7)

# Objectiu

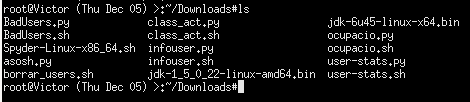
Ser capaços de programar tasques de manera puntual i periòdica.

# **Abans de començar**

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

* **ocupacio.sh**
* **badusers.py**

Comprovem si tenim els scripts:

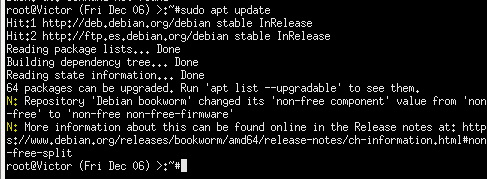


# **Tasques puntuals**

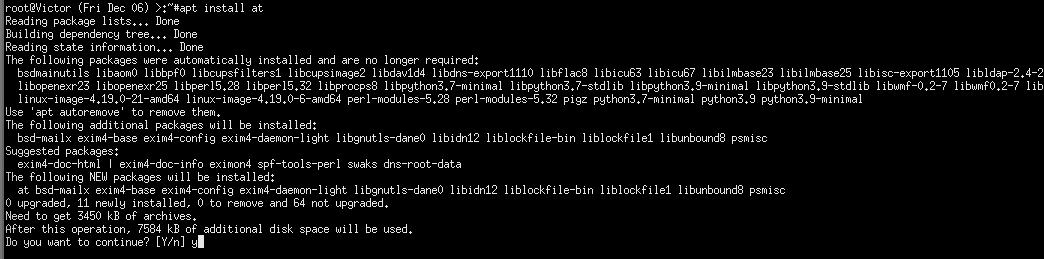
**Amb quina comanda es programen habitualment les tasques puntuals?**

Les tasques puntuals es programen amb la comanda . Per usar-la, hem de descargar-la.

**apt update**

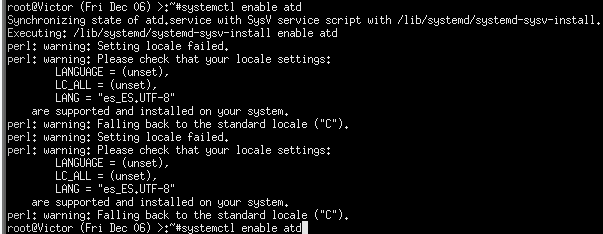


**apt install at**



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

**sudo systemctl enable atd**

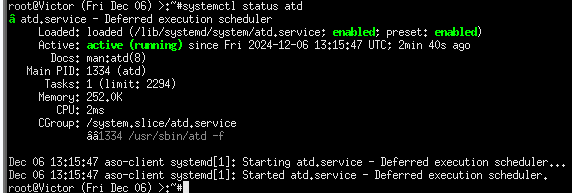
****

**sudo systemctl start atd**



Verifiquem l’estat

**sudo systemctl status atd**

****

**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**



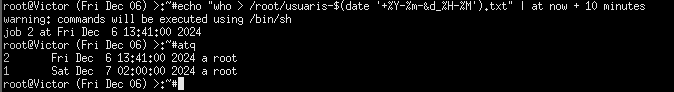
Si volem consultar les tasques que hem programat les podem veure amb la comanda atq.



**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**



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.**

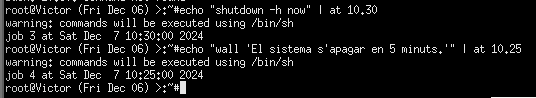
La comanda que utilitzarem és

**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:

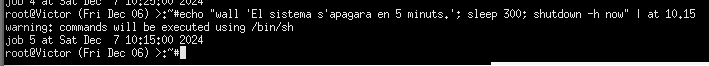
**echo “wall ‘El sistema s’apagará en 5 minuts.’” | at 10:25**

****

O també ho podem fer en una sola comanda

**echo "wall ‘El sistema s’apagará 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.



**Volem restringir aquest servei a l'usuari root.**

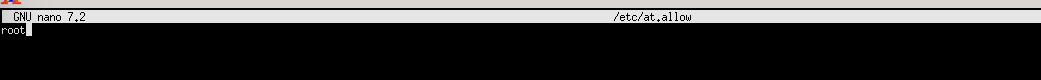
**Quin fitxer hem de crear? Amb quin contingut?**

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

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.

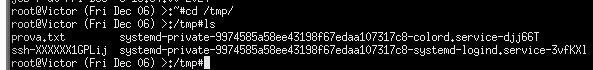
“At” mira primer el fitxer at.allow, i si existeix no mira el fitxer at.deny, per tant només hem de crear el fitxer at.allow i posar el nom de root.



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

Provem que funcioni amb root:







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



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

# Tasques periòdiques

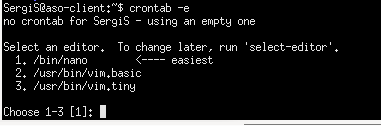
**Mireu el manual de les comandes cron i crontab i contesteu a les següents preguntes:**

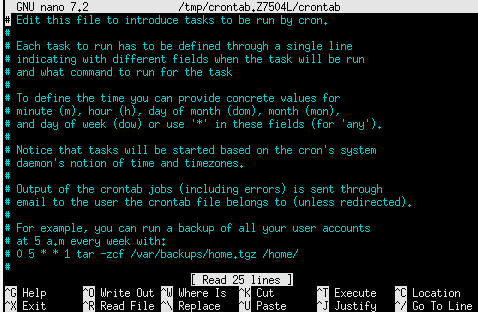
**Com podem veure quin es el contingut del cron d'un usuari?**

crontab -l

**Com podem afegir-hi noves tasques periòdiques?**

crontab -e





**Com podem limitar qui té accés al servei de cron?**

modifican els fitxers cron.allow i cron.deny

Un exemple a la hora de fer us d’aquestes comandes esmentades pot ser per permetre l’us unicament a l’usuari 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?**

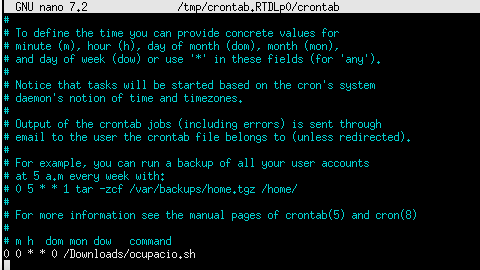
La validació immediata de les tasques cron pot prevenir errors de programació que podrien causar que les tasques no s'executin correctament. No obstant això, una validació automàtica o l'ús d'eines de verificació podria introduir una complexitat addicional o requerir permisos especials, i si la validació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

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

****

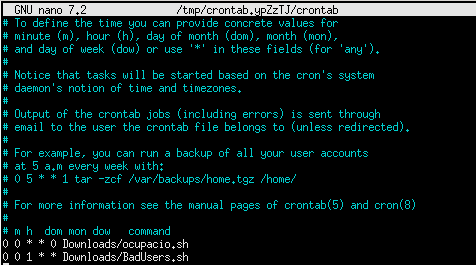
****

****

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

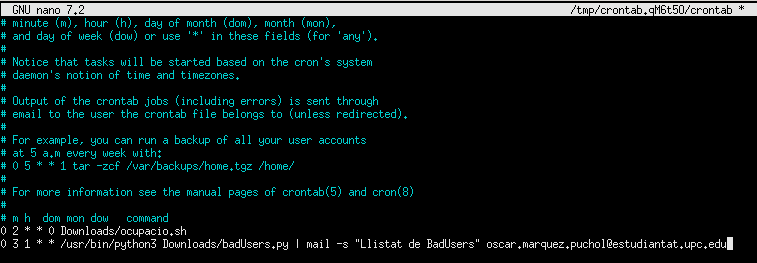
**Quina entrada afegireu al crontab? De quin usuari?**

****

****

****

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



**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.**

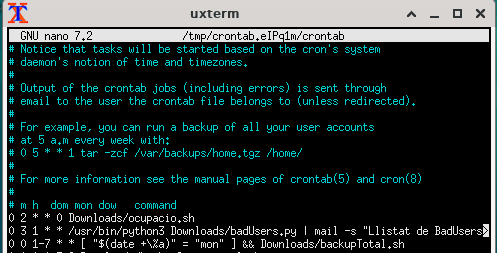
**El nom dels backups ha d'incloure la data en la que es varen fer i el tipus de backup.**

**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



#!/bin/bash

USER\_HOME="/home/sergiS"

BACKUP\_DIR="$USER\_HOME/backups"

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

BACKUP\_NAME="backup\_total\_$DATE.tar.gz"

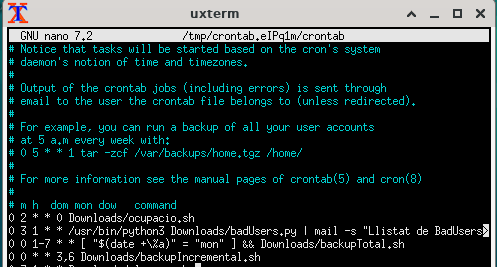
mkdir -p "$BACKUP\_DIR"

tar --exclude="$BACKUP\_DIR" -czf "$BACKUP\_DIR/$BACKUP\_NAME" "$USER\_HOME"

echo "Backup complet realitzat: $BACKUP\_NAME"



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



#!/bin/bash

# Variables

USER\_HOME="/home/sergiS"

BACKUP\_DIR="$USER\_HOME/backups"

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

BACKUP\_NAME="backup\_incremental\_$DATE.tar.gz"

SNAPSHOT\_FILE="$BACKUP\_DIR/snapshot.file"

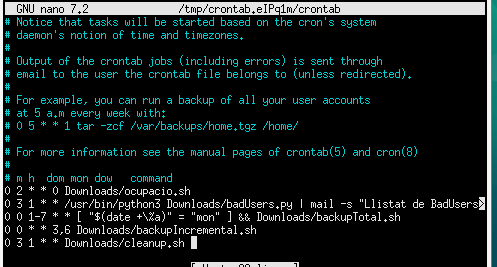
mkdir -p "$BACKUP\_DIR"

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).



#!/bin/bash

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

