

**ACTIVITAT AC9****Mòdul:** MP01 Sistemes Informàtics**UF:** UF2**Professor:** Marc Callejón**Data límit d'entrega:** 12/03/2025 23:59**Mètode d'entrega:** Per mitjà del Clickedu de l'assignatura.**Resultats de l'aprenentatge:** RA1. Administració de la Informació**Tarea 1 (1.5 punts): Monitorización del sistema**

1. Usa los siguientes comandos para obtener información del sistema y explica su

salida: -

uptime → Tiempo de actividad y carga del sistema.

```

11 de mar 10:06
arman@arman-VirtualBox: ~
arman@arman-VirtualBox:~$ uptime
10:04:43 up 2 min,  1 user,  load average: 6,25, 2,91, 1,10

```

- free -m → Estado de la memoria RAM.

```

arman@arman-VirtualBox:~$ free -m
              total        usado       libre  compartido   búf/caché  disponible
Mem:          5556          954        4078          33          779          4601
Inter:         4095           0        4095

```

- df -h → Uso de espacio en disco.

```

arman@arman-VirtualBox:~$ df -h
S.ficheros      Tamaño Usados  Disp Uso% Montado en
tmpfs           556M   1,5M   555M   1% /run
/dev/sda2       25G    11G   14G   44% /
tmpfs           2,8G     0   2,8G   0% /dev/shm
tmpfs           5,0M    8,0K   5,0M   1% /run/lock
tmpfs           556M   116K   556M   1% /run/user/1000
arman@arman-VirtualBox:~$

```

htop: Es una versión mejorada de `top`. Ofrece una interfaz más avanzado con colores. Además tiene mas detalles

3. Usa watch para ejecutar free -m cada 2 segundos y observar cómo cambia la memoria en tiempo real.

```
arman@arman-VirtualBox: ~  
Cada 2,0s: free -m  
arman-VirtualBox: Tue Mar 11 11:01:16 2025  
  
              total        usado        libre compartido      búf/caché  disponible  
Mem:          5556          1104          2983           33          1746          4451  
Inter:         4095           0          4095
```

Tarea 2 (1.5 puntos): Gestión de procesos

1. Ejecuta ps aux y responde:

```
arman@arman-VirtualBox: ~  
arman@arman-VirtualBox:~$ ps aux  
USER      PID %CPU %MEM    VSZ   RSS TTY      STAT START   TIME COMMAND  
root         1  0.2  0.2  23116 13900 ?        Ss   10:20   0:15 /sbin/init splash  
root         2  0.0  0.0      0     0 ?        S    10:20   0:01 [kthreadd]  
root         3  0.0  0.0      0     0 ?        S    10:20   0:00 [pool_workqueue_release]  
root         4  0.0  0.0      0     0 ?        I<   10:20   0:00 [kworker/R-rcu_g]  
root         5  0.0  0.0      0     0 ?        I<   10:20   0:00 [kworker/R-rcu_p]  
root         6  0.0  0.0      0     0 ?        I<   10:20   0:00 [kworker/R-slub_  
root         7  0.0  0.0      0     0 ?        I<   10:20   0:00 [kworker/R-netns]  
root        11  0.0  0.0      0     0 ?        I    10:20   0:00 [kworker/u16:0-ext4-rsv-conversion]  
root        12  0.0  0.0      0     0 ?        I<   10:20   0:00 [kworker/R-mm_pe]  
root        13  0.0  0.0      0     0 ?        I    10:20   0:00 [rcu_tasks_kthread]  
root        14  0.0  0.0      0     0 ?        I    10:20   0:00 [rcu_tasks_rude_kthread]  
root        15  0.0  0.0      0     0 ?        I    10:20   0:00 [rcu_tasks_trace_kthread]  
root        16  0.0  0.0      0     0 ?        S    10:20   0:04 [ksoftirqd/0]  
root        17  0.9  0.0      0     0 ?        I    10:20   0:49 [rcu_preempt]  
root        18  0.0  0.0      0     0 ?        S    10:20   0:01 [migration/0]  
root        19  0.0  0.0      0     0 ?        S    10:20   0:00 [idle_inject/0]  
root        20  0.0  0.0      0     0 ?        S    10:20   0:00 [cpuhp/0]  
root        21  0.0  0.0      0     0 ?        S    10:20   0:00 [cpuhp/1]
```

- ¿Cómo puedes filtrar los procesos de un usuario específico?

```
arman@arman-VirtualBox: ~  
arman@arman-VirtualBox:~$ ps aux | grep arman  
avahi       794  0.1  0.0   8664  4224 ?        Ss   10:20   0:02 avahi-daemon: running [arman-Vir  
ualBox.local]  
arman      1825  0.1  0.2  21120 13064 ?        Ss   10:21   0:04 /usr/lib/systemd/systemd --user  
arman      1831  0.0  0.0   21456  3736 ?        S    10:21   0:00 (sd-pam)  
arman      1843  0.0  0.2 124044 13860 ?        S<sl 10:21   0:01 /usr/bin/pipewire  
arman      1844  0.0  0.1 106404  6016 ?        Ssl  10:21   0:00 /usr/bin/pipewire -c filter-chat  
.conf  
arman      1845  0.1  0.2  39128 11648 ?        Ss   10:21   0:03 /snap/snapd-desktop-integration/  
53/usr/bin/snapd-desktop-integration  
arman      1848  0.0  0.3 415852 18816 ?        S<sl 10:21   0:01 /usr/bin/wireplumber  
arman      1850  0.0  0.2 124612 12796 ?        S<sl 10:21   0:00 /usr/bin/pipewire-pulse  
arman      1855  0.0  0.1 325400  9984 ?        SLsl 10:21   0:01 /usr/bin/gnome-keyring-daemon --  
foreground --components=pkcs11,secrets --control-directory=/run/user/1000/keyring  
arman      1858  0.5  0.1  10960  6784 ?        Ss   10:21   0:13 /usr/bin/dbus-daemon --session --  
address=systemd: --nofork --nopidfile --systemd-activation --syslog-only  
arman      1908  0.0  0.1 244664  6400 tty2      Ssl+ 10:21   0:00 /usr/libexec/gdm-wayland-session  
env GNOME_SHELL_SESSION_MODE=ubuntu /usr/bin/gnome-session --session=ubuntu  
arman      1914  0.0  0.2 307196 16768 tty2      Sl+   10:21   0:00 /usr/libexec/gnome-session-binar  
--session=ubuntu  
arman      1920  0.0  0.1 692996  7552 ?        Ssl  10:21   0:01 /usr/libexec/xdg-document-portal  
arman      1931  0.0  0.1 318408  6016 ?        Ssl  10:21   0:00 /usr/libexec/xdg-permission-stor  
arman      2012  0.0  0.1 162652  6784 ?        Ssl  10:21   0:00 /usr/libexec/gcr-ssh-agent --bas  
-dir /run/user/1000/gcr  
arman      2013  0.0  0.0 100216  5504 ?        Ssl  10:21   0:00 /usr/libexec/gnome-session-ctl -  
monitor
```

- ¿Cómo puedes listarsolo los procesos con mayor consumo de CPU?

```
arman@arman-VirtualBox:~$ ps aux --sort=-%cpu
USER      PID %CPU %MEM    VSZ   RSS TTY      STAT START   TIME COMMAND
arman    6658  100  0.0  22412  4608 pts/0    R+   11:33   0:00 ps aux --sort=-%cpu
arman    2074  19.9  6.7 5800416 386100 ?        Ssl  10:21  14:26 /usr/bin/gnome-shell
arman    6562  4.3  1.0 3024604 61532 ?        Sl   11:33   0:01 gjs /usr/share/gnome-shell/extensions/ding@rastersoft
root      17  1.0  0.0 0 0 ?        I    10:20   0:47 [rcu_preempt]
arman    4093  0.9  1.0 638856 59996 ?        Ssl  10:23   0:41 /usr/libexec/gnome-terminal-server
root     6421  0.6  0.7 587812 40508 ?        Ssl  11:18   0:06 /usr/libexec/fwupd/fwupd
root     6468  0.5  0.0 0 0 ?        I    11:18   0:04 [kworker/6:0-events]
arman    1858  0.3  0.1 11144 6784 ?        Ss   10:21   0:16 /usr/bin/dbus-daemon --session --address=systemd: --n
root      826  0.3  0.5 1616880 31736 ?        Ssl  10:20   0:16 /usr/lib/snapd/snapd
message+ 796  0.3  0.1 12144 7040 ?        Ss   10:20   0:16 @dbus-daemon --system --address=systemd: --nofork --n
arman    2380  0.3  0.5 430888 29544 ?        Sl   10:21   0:15 /usr/libexec/ibus-extension-gtk3
root      1  0.3  0.2 23116 13900 ?        Ss   10:20   0:15 /sbin/init splash
root    1479  0.2  0.1 325884 8832 ?        Ssl  10:21   0:12 /usr/libexec/upowerd
root     6267  0.2  0.0 0 0 ?        I    11:10   0:03 [kworker/6:2-cgroup_destroy]
polkitd   817  0.2  0.2 400132 12472 ?        Ssl  10:20   0:09 /usr/lib/polkit-1/polkitd --no-debug
root     6492  0.1  0.0 0 0 ?        I    11:23   0:01 [kworker/0:0-events]
arman    2233  0.1  0.2 324084 12180 ?        Ssl  10:21   0:07 /usr/bin/ibus-daemon --panel disable
root      54  0.1  0.0 0 0 ?        S    10:20   0:07 [ksoftirqd/6]
root     6198  0.1  0.0 0 0 ?        I    11:02   0:02 [kworker/0:1-mm_percpu_wq]
root     5909  0.1  0.0 0 0 ?        I    10:45   0:04 [kworker/6:1-events]
root      915  0.1  0.3 345672 19412 ?        Ssl  10:20   0:06 /usr/sbin/NetworkManager --no-daemon
root      336  0.1  0.3 51104 17408 ?        S<s  10:20   0:05 /usr/lib/systemd/systemd-journald
arman    1825  0.1  0.2 21128 13024 ?        Ss   10:21   0:05 /usr/lib/systemd/systemd --user
root     5930  0.1  0.0 0 0 ?        I    10:57   0:02 [kworker/3:0-events]
root     6287  0.1  0.0 0 0 ?        I<   11:18   0:01 [kworker/2:0H-kblockd]
root      24  0.1  0.0 0 0 ?        S    10:20   0:04 [ksoftirqd/1]
root     6276  0.1  0.0 0 0 ?        I    11:16   0:01 [kworker/2:1-mm_percpu_wq]
root     6472  0.0  0.0 0 0 ?        I<   11:20   0:00 [kworker/3:3H-kblockd]
root      16  0.0  0.0 0 0 ?        S    10:20   0:04 [ksoftirqd/0]
```

2. Usa kill para terminar un proceso de prueba (sleep 300 & para generar un PID).

```
arman@arman-VirtualBox:~$ sleep 300 &
[6] 6863
arman@arman-VirtualBox:~$ kill 6863
arman@arman-VirtualBox:~$
```

3. Cambia la prioridad de un proceso con renice.

```
arman@arman-VirtualBox:~$ sudo renice -n 10 -p 1931
1931 (process ID) prioridad anterior 0, nueva prioridad 10
arman@arman-VirtualBox:~$
```

4. Ejecuta un proceso en segundo plano (ping google.com > ping.log &) y tráelo al primer plano con fg.

```
arman@arman-VirtualBox:~$ ping google.com > ping.log &
[1] 7037
arman@arman-VirtualBox:~$ fg
ping google.com > ping.log
```

Tarea 3 (1.5 puntos): Automatización con Cron

1. Programa un cron que registre el uso de CPU cada 5 minutos en un archivo de log.

usar comando: **crontab -e**

```
arman@arman-VirtualBox: ~  
GNU nano 7.2 /tmp/crontab.oMon1R/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/  
#  
# For more information see the manual pages of crontab(5) and cron(8)  
#  
# m h dom mon dow command  
*/5 * * * * top -bn1 | grep "Cpu(s)" >> /home/arman/logfile.log
```

```
arman@arman-VirtualBox:~$ crontab -e  
crontab: installing new crontab  
arman@arman-VirtualBox:~$
```

2. Configura un cron para borrar archivostemporales en /tmp cada día a las 4 AM.

usar comando: **crontab -e**


```
arman@arman-VirtualBox: ~
GNU nano 7.2 /tmp/crontab.dUWGHj/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/
#
# For more information see the manual pages of crontab(5) and cron(8)
#
# m h dom mon dow  command

*/5 * * * * top -bn1 | grep "Cpu(s)" >> /home/arman/logfile.log

0 4 * * * rm -rf /tmp/*
```

3. Modifica /etc/crontab para ejecutar un script solo los lunes y miércoles.

crear un archivo script y luego usa el comando sudo nano /etc/crontab

```
12 de mar 12:11
arman@arman-VirtualBox: ~
GNU nano 7.2 /etc/crontab *

# /etc/crontab: system-wide crontab
# Unlike any other crontab you don't have to run the `crontab'
# command to install the new version when you edit this file
# and files in /etc/cron.d. These files also have username fields,
# that none of the other crontabs do.

SHELL=/bin/sh
# You can also override PATH, but by default, newer versions inherit it from the environment
#PATH=/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin

# Example of job definition:
# .----- minute (0 - 59)
# | .----- hour (0 - 23)
# | | .----- day of month (1 - 31)
# | | | .----- month (1 - 12) OR jan,feb,mar,apr ...
# | | | | .---- day of week (0 - 6) (Sunday=0 or 7) OR sun,mon,tue,wed,thu,fri,sat
# | | | | |
# * * * * * user-name command to be executed
17 * * * * root cd / && run-parts --report /etc/cron.hourly
25 6 * * * root test -x /usr/sbin/anacron || { cd / && run-parts --report /etc/cron.daily
47 6 * * 7 root test -x /usr/sbin/anacron || { cd / && run-parts --report /etc/cron.weekly
52 6 1 * * root test -x /usr/sbin/anacron || { cd / && run-parts --report /etc/cron.monthly
#
0 0 * * 1,3 /home/arman/scripts/script.sh
```

Tarea 4 (1.5 punts): Monitorización de logs y usuarios

1. Usa `tail -f /var/log/syslog` y observa cómo los registros del sistema se actualizan en tiempo real.

```
arman@arman-VirtualBox: ~  
arman@arman-VirtualBox:~$ tail -f /var/log/syslog  
2025-03-12T12:12:24.922572+01:00 arman-VirtualBox apt.systemd.daily[3872]: /usr/bin/unattended-upgrade:567: DeprecationW  
arning: This process (pid=3872) is multi-threaded, use of fork() may lead to deadlocks in the child.  
2025-03-12T12:12:24.922646+01:00 arman-VirtualBox apt.systemd.daily[3872]: pid = os.fork()  
2025-03-12T12:12:29.663943+01:00 arman-VirtualBox apt.systemd.daily[3872]: /usr/bin/unattended-upgrade:567: DeprecationW  
arning: This process (pid=3872) is multi-threaded, use of fork() may lead to deadlocks in the child.  
2025-03-12T12:12:29.664003+01:00 arman-VirtualBox apt.systemd.daily[3872]: pid = os.fork()  
2025-03-12T12:12:37.637967+01:00 arman-VirtualBox systemd[1]: Reexecuting requested from client PID 5600 ('systemctl') (u  
nit apt-daily-upgrade.service)...  
2025-03-12T12:12:37.645295+01:00 arman-VirtualBox systemd[1]: Reexecuting.  
2025-03-12T12:12:37.804534+01:00 arman-VirtualBox kernel: systemd[1]: systemd 255.4-1ubuntu8.4 running in system mode (+  
PAM +AUDIT +SELINUX +APPARMOR +IMA +SMACK +SECCOMP +GCRYPT -GNUTLS +OPENSSL +ACL +BLKID +CURL +ELFUTILS +FIDO2 +IDN2 -ID  
N +IPTC +KMOD +LIBCRYPTSETUP +LIBFDISK +PCRE2 -PWQUALITY +P11KIT +QRENCODE +TPM2 +BZIP2 +LZ4 +XZ +ZLIB +ZSTD -BPF_FRAMEW  
ORK -XKBCOMMON +UTMP +SYSVINIT default-hierarchy=unified)  
2025-03-12T12:12:37.804556+01:00 arman-VirtualBox kernel: systemd[1]: Detected virtualization oracle.  
2025-03-12T12:12:37.804558+01:00 arman-VirtualBox kernel: systemd[1]: Detected architecture x86-64.  
2025-03-12T12:12:37.966201+01:00 arman-VirtualBox kernel: systemd[1]: Configuration file /run/systemd/system/netplan-ovs  
-cleanup.service is marked world-inaccessible. This has no effect as configuration data is accessible via APIs without r  
estrictions. Proceeding anyway.  
2025-03-12T12:13:01.443782+01:00 arman-VirtualBox cron[862]: (*system*) RELOAD (/etc/crontab)  
2025-03-12T12:13:01.444495+01:00 arman-VirtualBox cron[862]: Error: bad command; while reading /etc/crontab  
2025-03-12T12:13:01.444536+01:00 arman-VirtualBox cron[862]: (*system*) ERROR (Syntax error, this crontab file will be i  
gnored)  
2025-03-12T12:13:29.105670+01:00 arman-VirtualBox apt.systemd.daily[3872]: /usr/bin/unattended-upgrade:567: DeprecationW  
arning: This process (pid=3872) is multi-threaded, use of fork() may lead to deadlocks in the child.  
2025-03-12T12:13:29.106755+01:00 arman-VirtualBox apt.systemd.daily[3872]: pid = os.fork()  
2025-03-12T12:13:35.194381+01:00 arman-VirtualBox apt.systemd.daily[3872]: /usr/bin/unattended-upgrade:567: DeprecationW  
arning: This process (pid=3872) is multi-threaded, use of fork() may lead to deadlocks in the child.
```

2. Ejecuta `who` y `w` para ver qué usuarios están conectados y qué procesos están ejecutando.

```
arman@arman-VirtualBox:~$ who  
arman    seat0      2025-03-12 11:37 (login screen)  
arman    tty2       2025-03-12 11:37 (tty2)  
arman@arman-VirtualBox:~$ w  
 12:15:09 up 39 min,  1 user,  load average: 2,36, 1,88, 1,16  
USUARIO  TTY      DESDE          LOGIN@  IDLE   JCPU   PCPU   WHAT  
arman    tty2     -              11:37   39:12  0.17s  0.13s  /usr/libexec/gnome-sessio  
arman@arman-VirtualBox:~$
```

3. Usa last para ver el historial de sesiones iniciadas en el sistema.

```
arman@arman-VirtualBox:~$ last
arman    tty2          tty2          Wed Mar 12 11:37    still logged in
arman    seat0         login screen  Wed Mar 12 11:37    still logged in
reboot   system boot   6.8.0-49-generic Wed Mar 12 11:35    still running
arman    tty2          tty2          Wed Mar 12 11:32    - crash (00:03)
arman    seat0         login screen  Wed Mar 12 11:32    - crash (00:03)
reboot   system boot   6.8.0-49-generic Wed Mar 12 11:29    still running
arman    tty2          tty2          Tue Mar 11 10:21    - crash (1+01:08)
arman    seat0         login screen  Tue Mar 11 10:21    - crash (1+01:08)
reboot   system boot   6.8.0-49-generic Tue Mar 11 10:20    still running
arman    tty2          tty2          Tue Mar 11 10:16    - crash (00:03)
arman    seat0         login screen  Tue Mar 11 10:16    - crash (00:03)
reboot   system boot   6.8.0-49-generic Tue Mar 11 10:15    still running
reboot   system boot   6.8.0-49-generic Tue Mar 11 10:14    still running
arman    tty2          tty2          Tue Mar 11 10:03    - crash (00:10)
arman    seat0         login screen  Tue Mar 11 10:03    - crash (00:10)
reboot   system boot   6.8.0-49-generic Tue Mar 11 10:02    still running
arman    tty2          tty2          Tue Feb 25 10:30    - crash (13+23:32)
arman    seat0         login screen  Tue Feb 25 10:30    - crash (13+23:32)
reboot   system boot   6.8.0-49-generic Tue Feb 25 10:24    still running
arman    tty2          tty2          Sun Dec 8 22:30     - crash (78+11:54)
arman    seat0         login screen  Sun Dec 8 22:30     - crash (78+11:54)
reboot   system boot   6.8.0-49-generic Sun Dec 8 22:29     still running
arman    tty2          tty2          Thu Dec 5 10:35     - crash (3+11:54)
arman    seat0         login screen  Thu Dec 5 10:35     - crash (3+11:54)
reboot   system boot   6.8.0-49-generic Thu Dec 5 10:31     still running
reboot   system boot   6.8.0-49-generic Thu Dec 5 10:22     still running
```

Tarea 5 (4 puntos) CONFIGURACIÓN DE RED Y ÓRDENES DE CONSOLA

Configura una red local entre al menos dos o más máquinas virtuales con Ubuntu (Será necesario que busques información sobre la configuración con adaptador puente en un caso real o red interna para virtualización).

Pues cada maquina configura los Ips en una red . Esto se puede hacer con configuracion o con terminal

Configuración

Dar clic al icono del configuración de la red ahí le das aip manual ahí los pones el por ejemplo 192.168..1.1 y con mascara y la otra maquina puede ser 192.168.1.2 y con la mascara ademas en DNS pon la ip de primer ordenador que es 192.168.1.1 y dar a aplicar. ahí puede ser 2 opciones de dns tambien puedes poner 8.8.8.8

Terminal

Tenies que configurar el archivo /etc/netplan/00-installer-config.yaml : Se puede utilizar el comando sudo nano /etc/netplan/00-installer-config.yaml

ahi dentro lo tienes que configurar des de principio algo asi:

Ejemplo

network:

version: 2

renderer: networkd

ethernets:

enp0s3:

dhcp4: no

addresses:

- 192.168.1.1/24

gateway4: 192.168.1.1

nameservers:

addresses:

8.8.8.8

Luego de editar el archivo tienes que escribir el comando

sudo netplan apply

Asi te lo configura

- Utiliza el comando adecuado para verificar la dirección IP asignada a cada ordenador de la red.

El comando es `ip a` o `ifconfig`

- Utiliza la herramienta `ping` para verificar la conectividad entre los ordenadores de la red.
- Utiliza la herramienta `nmap` para escanear la red y determinar qué dispositivos están conectados a ella.

para hacer el ping pon el comando `ping` y ip del otro ordenador como si estoy en ordenador 1 pues pongo de ordenador 2 comando: `ping 192.168.1.2` y lo mismo al revés

y para saber los dispositivos conectados ponemos usamos `nmap`:

`sudo nmap -sP 192.168.1.0/24`

- Utiliza la herramienta `traceroute` para determinar la ruta que sigue un paquete de datos desde uno de los ordenadores de la red hasta otro.

el comando de `tracert` es :

`sudo traceroute 192.168.1.1` o `192.168.1.2` dependiendo de la máquina en que estais

- Utiliza la herramienta `netstat` para examinar las conexiones de red y las tablas de enrutamiento.

comando es : `netstat -tuln`

y para enrutamiento: `netstat -r`

- Utiliza la herramienta `iftop` para monitorizar el tráfico de red en tiempo real.
- Verifica la velocidad de la conexión a Internet utilizando la herramienta `speedtest-cli`.

--