

## Est.: Meda Margueiz, Christian Eduardo-MM17017

### Actividad I

Describe el funcionamiento de los siguientes comando y ejemplifique un escenario para su uso

htop: administra interactivamente el sistema

\$htop

```
htop
Descargas: apt-get

1  [|||||] 13.2% Tasks: 129, 604 thr; 3 running
2  [|||||] 22.5% Load average: 0.58 0.91 0.80
3  [|||||] 14.8% Uptime: 07:17:42
4  [|||||] 13.5%
Mem[|||||] 3.11G/7.73G
Swp[|||||] 0K/1.77G

PID USER PRI NI VIRT RES SHR S CPU% MEM% TIME+ Command
15895 cmargueiz 20 0 2880M 258M 133M S 19.5 3.3 0:30.87 /usr/lib/firefox/firefox -contentproc -childID 27
11798 cmargueiz 20 0 2888M 232M 109M S 13.4 2.9 0:44.61 /usr/lib/firefox/firefox -contentproc -childID 18
896 root 20 0 549M 84652 55828 S 7.4 1.0 13:47.83 /usr/lib/xorg/Xorg -core :0 -seat seat0 -auth /va
11987 cmargueiz 20 0 2525M 156M 58068 S 5.4 2.0 2:49.66 io.elementary.music
1520 cmargueiz 20 0 1633M 114M 59204 R 5.4 1.4 10:13.50 gala
3102 cmargueiz 20 0 4089M 551M 178M S 4.7 7.0 45:04.12 /usr/lib/firefox/firefox
12005 cmargueiz 20 0 2525M 156M 58068 S 2.7 2.0 0:18.87 io.elementary.music
16416 cmargueiz 20 0 567M 32884 26564 S 2.0 0.4 0:00.73 io.elementary.screenshot-tool
16400 cmargueiz 20 0 59448 5124 3932 R 2.0 0.1 0:00.47 htop
15898 cmargueiz 20 0 2880M 258M 133M S 2.0 3.3 0:02.26 /usr/lib/firefox/firefox -contentproc -childID 27
13558 cmargueiz 20 0 696M 46484 31996 S 2.0 0.6 0:06.88 io.elementary.terminal
16286 cmargueiz 20 0 2525M 156M 58068 S 1.3 2.0 0:02.34 io.elementary.music
3113 cmargueiz 20 0 4089M 551M 178M S 1.3 7.0 6:47.00 /usr/lib/firefox/firefox
1490 cmargueiz 9 -11 3630M 20944 16480 S 1.3 0.3 3:32.10 /usr/bin/pulseaudio --start --log-target=syslog
927 root 20 0 549M 84652 55828 S 1.3 1.0 0:43.94 /usr/lib/xorg/Xorg -core :0 -seat seat0 -auth /va
785 root 20 0 263M 6136 5276 S 1.3 0.1 3:08.90 /usr/sbin/iio-sensor-proxy
15974 cmargueiz 20 0 1347M 227M 111M S 1.3 2.9 0:20.97 /usr/share/discord/Discord --type=renderer --auto
415 root 19 -1 364M 208M 207M S 1.3 2.6 1:18.15 /lib/systemd/systemd-journald
3137 cmargueiz 20 0 4089M 551M 178M S 0.7 7.0 3:56.84 /usr/lib/firefox/firefox
1541 cmargueiz 20 0 699M 59684 27652 S 0.7 0.7 0:34.48 plank
923 root 20 0 549M 84652 55828 S 0.7 1.0 1:03.74 /usr/lib/xorg/Xorg -core :0 -seat seat0 -auth /va
3228 cmargueiz 20 0 3685M 643M 179M S 0.7 8.1 18:23.42 /usr/lib/firefox/firefox -contentproc -childID 1
F1Help F2Setup F3Search F4Filter F5Tree F6SortBy F7Nice F8Nice F9Kill F10Quit
```

top: Muestra los procesos de linux

```
top
top - 01:22:43 up 7:20, 1 user, load average: 0.70, 0.92, 0.83
Tareas: 238 total, 1 ejecutar, 168 hibernar, 2 detener, 0 zombie
%Cpu(s): 8.0 usuario, 3.1 sist, 0.0 adecuado, 88.7 inact, 0.2 en espera, 0.0 hardw int, 0.1 softw int, 0.
KiB Mem : 8101628 total, 685972 libre, 3107028 usado, 4308628 búfer/caché
KiB Intercambio: 1852220 total, 1852220 libre, 0 usado. 4580004 dispon Mem

PID USUARIO PR NI VIRT RES SHR S %CPU %MEM HORA+ ORDEN
16179 cmargue+ 20 0 2785056 259300 131460 S 15.2 3.2 0:28.48 Web Content
3102 cmargue+ 20 0 4191596 575812 182576 S 6.0 7.1 45:24.39 firefox
11987 cmargue+ 20 0 2585628 160148 58068 S 5.6 2.0 2:58.67 io.elementary.m
896 root 20 0 563576 84992 55920 S 4.0 1.0 13:59.15 Xorg
1520 cmargue+ 20 0 1672244 117292 59204 S 2.3 1.4 10:21.17 gala
1490 cmargue+ 9 -11 3717940 21316 16852 S 2.0 0.3 3:35.16 pulseaudio
3228 cmargue+ 20 0 3773452 650756 183948 S 1.3 8.0 18:25.02 Web Content
16595 cmargue+ 20 0 581108 31732 25536 S 1.3 0.4 0:00.35 io.elementary.s
13558 cmargue+ 20 0 712844 46484 31996 S 1.0 0.6 0:09.29 io.elementary.t
785 root 20 0 269708 6136 5276 S 0.7 0.1 3:10.22 iio-sensor-prox
11432 cmargue+ 20 0 2892992 233200 140244 S 0.7 2.9 4:54.33 Web Content
13316 cmargue+ 20 0 3123644 409072 110704 S 0.7 5.0 1:05.49 Web Content
15974 cmargue+ 20 0 1374376 230644 110216 S 0.7 2.8 0:23.01 Discord
16581 cmargue+ 20 0 69944 4128 3476 R 0.7 0.1 0:00.12 top
10 root 20 0 0 0 0 I 0.3 0.0 0:24.83 rcu_sched
415 root 19 -1 381756 217672 216492 S 0.3 2.7 1:18.92 systemd-journal
574 root -51 0 0 0 0 S 0.3 0.0 1:37.58 irq/40-iwlwifi
838 syslog 20 0 263032 4340 3724 S 0.3 0.1 0:15.01 rsyslogd
1362 cmargue+ 20 0 50052 4300 3836 S 0.3 0.1 0:00.78 dbus-daemon
1541 cmargue+ 20 0 715864 59684 27652 S 0.3 0.7 0:34.92 plank
1557 cmargue+ 20 0 499720 50448 20124 S 0.3 0.6 0:23.44 bamfdaemon
11158 root 20 0 0 0 0 I 0.3 0.0 0:00.74 kworker/2:2-eve
1 root 20 0 225340 9084 6768 S 0.0 0.1 0:02.41 systemd
2 root 20 0 0 0 0 S 0.0 0.0 0:00.02 kthreadd
3 root 0 -20 0 0 0 I 0.0 0.0 0:00.00 rcu_gp
```

\$Stop

awk: es una herramienta de procesamiento de patrones en líneas de texto

Imprimir último campo de cada línea:

```
# awk '{ print $NF }' fichero
```

ps: informa instantaneamente los procesos actuales.

```
$ps
```

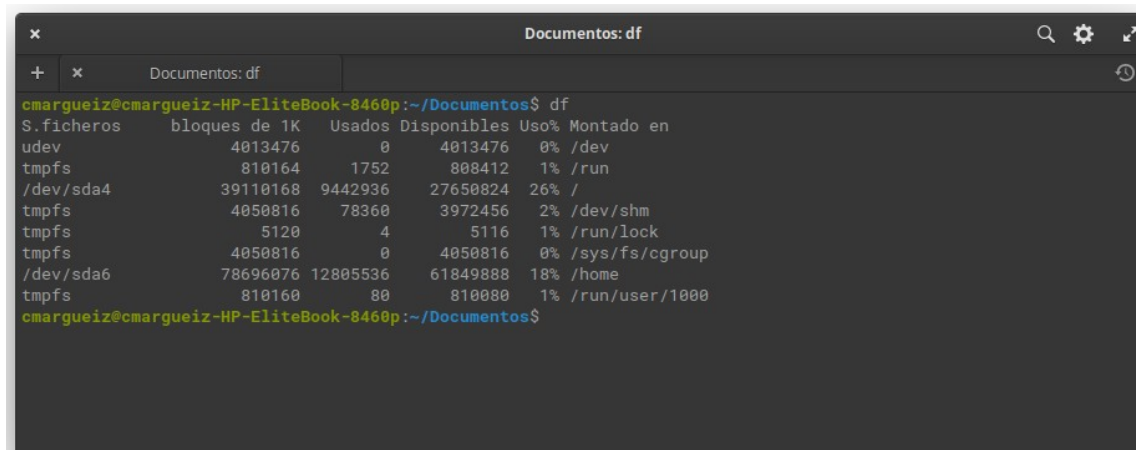
kill: es un comando que envía una señal de terminación

```
$kill oneko
```

killall: es que puedes matar cualquier proceso con solo el nombre del comando. Si más de un proceso está ejecutando el comando especificado, matará a todos.

```
#killall -v apache2
```

df: es una herramienta CLI del tipo Unix que nos permite conocer la cantidad de espacio libre y espacio utilizado por nuestro sistema de archivos en nuestras unidades de almacenamiento



```
cmargueiz@cmargueiz-HP-EliteBook-8460p:~/Documentos$ df
```

S.ficheros	bloques de 1K	Usados	Disponibles	Uso%	Montado en
udev	4013476	0	4013476	0%	/dev
tmpfs	810164	1752	808412	1%	/run
/dev/sda4	39110168	9442936	27650824	26%	/
tmpfs	4050816	78360	3972456	2%	/dev/shm
tmpfs	5120	4	5116	1%	/run/lock
tmpfs	4050816	0	4050816	0%	/sys/fs/cgroup
/dev/sda6	78696076	12805536	61849888	18%	/home
tmpfs	810160	80	810080	1%	/run/user/1000

```
cmargueiz@cmargueiz-HP-EliteBook-8460p:~/Documentos$
```

free: Es un comando que puede en GNU/Linux resultarnos muy útil a la hora de entender el consumo de nuestra memoria

```
[cristian@ARCH ~]$ free
              total        used        free      shared  buff/cache   available
Mem:          10178092     2730420     3956260      188824      3491412      6946156
Swap:          1057236           0      1057236
```

## Actividad II

Explique el funcionamiento de los siguientes comandos.

Htop: muestra el uso de la memoria y los procesos que ocurren

top: Muestra los diferentes procesos del sistema linux

top -n 1: Especifica el número máximo de iteraciones, o cuadros, que top debería producir antes de finalizar en este caso es una iteración.

ps -aux: Lista los procesos de todos los usuarios con información añadida

ps -aux | grep "usuario": Lista los procesos de todos los usuarios con información añadida según el nombre del usuario.

killall -u "usuario" (si se usa como root, colgaría el sistema): Mata solo los procesos de los derechos del usuario especificado.

foo="Hola": Asigna la cadena "Hola" a la variable de entorno \$foo

foo="\$foo Mundo": Concatena la cadena de texto " Mundo" al contenido de la variable foo

echo \$foo: Imprime en consola "Hola mundo"

### Actividad III

- Describa los tipos de rsyslog security levels con sus codigos, sus rsyslog y su severidad

Codigo	rsyslog	severidad
0	emerg, panic	Emergencia: el sistema no se puede usar
1	alert	Alerta: se deben tomar medidas de inmediato
2	crit	Crítico: condiciones críticas
3	err, error	Error: condiciones de error
4	warning, warn	Advertencia: condiciones de advertencia
5	notice	Aviso: condición normal pero significativa
6	info	Informativo: mensajes informativos
7	debug	Depuración: mensajes de nivel de depuración

- Defina el funcionamiento de journalctl

Permite acceder y manipular los datos del registro de los logs del sistema, puedes ver todos los datos o especificarlos y tambien decidir en que formato lo quieres como Json o otro

- Explique el funcionamiento de los siguientes comandos

**service rsyslog restart:**reinicia el servicio de rsyslog y aplica los cambios hechos en configuración si es que se hizo, pero interrumpiendo todas las conexiones establecidas en ese momento

**systemctl restart rsyslog.service:** reinicia el servicio de rsyslog (hace lo mismo que el comando anterior)

**sudo journalctl:** muestralos registros de los logs del sistema

**journalctl -r:** Nmuestra los registros de los logs del sistema en orden invertido, los primeros debajo, y los últimos encima.

- Describa que hacen las siguientes acciones cuando se están visualizando los logs del sistema

**down arrow key, enter, e, or j:** muestra el siguiente logs de la lista subiendo en la pantalla uno por uno y desapareciendo el que esta al principio de la panalla

**up arrow key, y, or k:** sube en la lista haciendo que desapareca uno por uno los de abajo y que aparescan los que estaban arriban en la pantalla es decir los que estaban al principio de la lista

**space bar:** hace un salto de pagina en la lista de logs subiendo todos los que se mostraban en pantalla y mostrando uno nuevos que estaban mas abajo en la lista de logs

**b:** hace un salto de pagina en la lista de logs bajando todos los que se mostraban en pantalla y mostrando los que estaban arriba de los que se mostraban en pantalla

**/search term:** Busca hacia adelante desde la posicion actual para la cadena de terminos relacionados a la busqueda

**?search term:** Busca hacia atras desde la posicion actual para la cadena de terminos relacionados a la busqueda

**'<c>:** Regresa a una marca, donde <c> es la etiqueta de carácter individual para la marca

**q:** se sale de vista de los registros de logs

- ¿Qué comando debe de utilizar cuando se ven los journal log del sistema y desea filtrarlos por un rango de fechas específicas?

`sudo journalctl --since «YYYY-MM-DD HH:MM:SS» --until «YYYY-MM-DD HH:MM:SS»`

- ¿Qué comando se puede ejecutar si deseo saber mediante los journal logs la lista de los boots disponibles?

`journalctl --list-boots` (lista de todos los boot)

`journalctl -b` (para ver los log boot actuales)

- ¿Con cuál comando se pueden ver los mensajes del kernel?

`journalctl -k`

- Describa la función de los diferentes formatos presentados a continuación mientras se observan los journal logs del sistema

**short:** es el valor predeterminado y genera una salida que es casi idéntica al formato de los archivos clásicos de syslog

**verbose:** muestra los elementos de entrada completamente estructurados con todos los campos

**json:** formatea las entradas como estructuras de datos JSON, una por línea

**json-pretty:** formatea las entradas como estructuras de datos JSON, pero las formatea en varias líneas para que sean más legibles para los humanos

**Cat:** genera una salida muy concisa, que solo muestra el mensaje real de cada entrada de diario sin metadatos, ni siquiera una marca de tiempo

(nota para cambiar el formato se utiliza el siguiente comando `journalctl -o formato` a usar)