# PRÁCTICA PARCIAL 2 DE ADMINISTRACIÓN DE SISTEMAS

Obtención diaria de estadísticas

Álvaro García García DNI:4138930G

# **FICHEROS**

#### tiemUsuarConec.sh:

Este script se encarga de almacenar en un fichero el tiempo total por día que ha habido gente conectada al sistema.

```
#!/bin/bash
echo "ESTADISTICAS REALIZADAS EN EL SIGUIENTE DIA Y HORA: " >> /root/.estadisticas/tiemUsuarConec.t
date +"%H:%M" >> /root/.estadisticas/tiemUsuarConec.txt
ac -d |sort -n -k4 -r |head -n 20 >> /root/.estadisticas/tiemUsuarConec.txt
```

- ac-> para obtener las estadísticas del tiempo de conexión
- -d-> para obtener el tiempo de conexión medido por días

**sort**-> lo utilizamos junto con el -k4 para ordenar tomando como referencia la cuarta columna, con -n tomamos los valores de tiempo y los interpretamos como números y con el -r lo ordenamos de mayor a menor.

**head->** tomamos las líneas desde la "cabeza" es decir desde arriba del todo y en concreto, tomamos 20 líneas

```
Apr 25 total 13.54
Apr 23 total 11.60
Mar 19 total 10.63
Apr 12 total 5.40
Today total 3.09
ESTADISTICAS REALIZADAS EN EL SIGUIENTE DIA Y HORA:
26/04/2020
21:54
Mar 20 total 13.84
Apr 25 total 13.64
Apr 27 total 13.84
Apr 28 total 11.60
Mar 19 total 5.40
Today total 3.11
ESTADISTICAS REALIZADAS EN EL SIGUIENTE DIA Y HORA:
26/04/2020
21:55
Mar 20 total 13.84
Apr 25 total 13.64
Apr 27 total 5.40
Today total 5.40
Today total 3.11
ESTADISTICAS REALIZADAS EN EL SIGUIENTE DIA Y HORA:
26/04/2020
21:55
Mar 20 total 14.47
Apr 7 total 13.84
Apr 25 total 13.64
Apr 25 total 13.54
Apr 27 total 13.84
Apr 28 total 11.60
Mar 19 total 10.63
Apr 12 total 6.77
Apr 6 total 5.40
Today total 3.12

© Ver ayuda 0 Guardar Mar Buscar Cortan txt J Justificar Posición New Deshacer
X Salir R Leer fich. Neemplazar V Pegar txt T Ortografía I r a línea New Rehacer
```

Se puede ver en el fichero "txt" como el tiempo de "today" incrementa un poco cada minuto.

#### tiemCadaUsuarConec.sh:

Cada vez que se ejecute, almacenará los 20 usuarios que más tiempo han estado conectados al sistema.

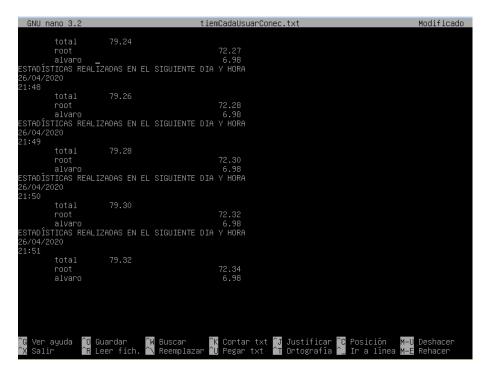
```
#!/bin/bash

echo "ESTADÍSTICAS REALIZADAS EN EL SIGUIENTE DIA Y HORA" >> /root/.estadisticas/tiemCadaUsuarConec
date +"%d/%m/%Y" >> /root/.estadisticas/tiemCadaUsuarConec.txt
date +"%H:%M" >> /root/.estadisticas/tiemCadaUsuarConec.txt
ac -p |sort -n -k2 -r |head -n 20 >> /root/.estadisticas/tiemCadaUsuarConec.txt
```

- ac-> para obtener las estadísticas del tiempo de conexión
- -p-> para obtener el tiempo de conexión total de cada usuario

**sort->** lo utilizamos junto con el -k2 para ordenar tomando como referencia la segunda columna, con -n tomamos los valores de tiempo y los interpretamos como números y con el -r lo ordenamos de mayor a menor.

**head->** tomamos las líenas desde la "cabeza" es decir desde arriba del todo y en concreto, tomamos 20 lineas



Se puede ver como el tiempo del root (que es el que estuve utilizando) incrementa un poco cada minuto.

### commandCPU.sh:

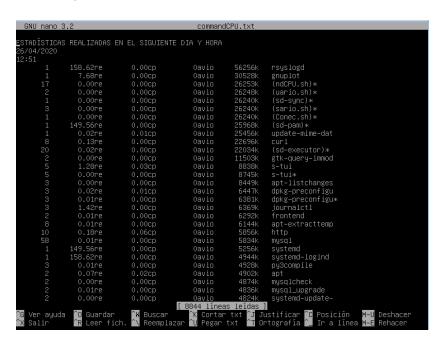
Se encarga de almacenar los 30 comandos que más CPU han consumido del sistema.

```
#!/bin/bash

echo "ESTADÍSTICAS REALIZADAS EN EL SIGUIENTE DIA Y HORA" >> /root/.estadisticas/commandCPU.txt
date +"%d/%m/%Y" >>/root/.estadisticas/commandCPU.txt
date +"%H:%M" >>/root/.estadisticas/commandCPU.txt
sa -a | sort -k5 -r | head -n 30 >> /root/.estadisticas/commandCPU.txt
```

- sa-> muestra las estadísticas de ejecución de distintos programas
- **-a**-> Muestra todos los programas
- **sort**-> lo utilizamos junto con el -k5 para ordenar tomando como referencia la quinta columna, y con el -r lo ordenamos de mayor a menor.

**head**-> tomamos las líneas desde la "cabeza" es decir desde arriba del todo y en concreto, tomamos 30 líneas.



Se puede ver como los resultados aparecen ordenados por consumo de CPU.

#### comandosMasEjecutados.sh:

Muestra los 30 comandos que más veces se han ejecutado en el sistema.

```
#!/bin/bash

echo "INFORMACION ALMACENADA A DIA Y HORA:" >> /root/.estadisticas/comandosMasEjecutados.txt
date +"%d/%m/%Y" >> /root/.estadisticas/comandosMasEjecutados.txt
date +"%H:%M" >> /root/.estadisticas/comandosMasEjecutados.txt
sa |sort -k1 -r |head -n 30 >> /root/.estadisticas/comandosMasEjecutados.txt
```

sa-> muestra las estadísticas de ejecución de distintos programas

**sort**-> lo utilizamos junto con el -k1 para ordenar tomando como referencia la primera columna, y con el -r lo ordenamos de mayor a menor.

**head**-> tomamos las líneas desde la "cabeza" es decir desde arriba del todo y en concreto, tomamos 30 líneas

GNU nano	3.2		comandosMasEje	ecutados.t	txt
INFORMACION 26/04/2020	I ALMACENADA A	A DIA Y HORA:			
12:51					
6811	3925.98re	0.67cp	0avio	2068k	
644	0.52re	0.08cp	0avio	2648k	dpkg-deb*
551	0.01re	0.00cp	0avio	2066k	bash*
464	0.00re	0.00cp	0avio	2426k	mandb*
438	0.01re	0.00cp	0avio	1391k	head
414	0.01re	0.00cp	0avio	1446k	sort

0	GNU nano 3.2			comandosMasEjecutados.txt				
		0.01re ALMACENADA	0.00cp A DIA Y HORA:	0avio	3854k	apt-config		
	/04/2020 :52 6837	3925.98re	0.67cp	0avio	2065k			
	644	0.52re	0.08cp	Oavio	2648k	dpkg-deb*		
	551	0.01re	0.00cp	Oavio	2066k	bash*		
	464	0.00re	0.00cp	Oavio	2426k	mandb*		
	443	0.01re	0.00cp	Oavio	1391k	head		
	419	0.01re	0.00cp	0avio	1445k	sort		

Podemos ver como el número de veces ejecutadas de head o sort incrementa en cada minuto.

#### numeroComandosPorUsuario.sh

Nos almacenará en un fichero el número total de comandos que ha ejecutado cada usuario en el sistema.

```
#!/bin/bash

echo "ESTADÍSTICAS OBTENIDAS A DÍA Y HORA: ">> /root/.estadisticas/numeroComandosPorUsuario.txt
date +"%d/%m/%Y" >> /root/.estadisticas/numeroComandosPorUsuario.txt
date +"%H:%M" >> /root/.estadisticas/numeroComandosPorUsuario.txt
sa -m |sort -k1 -r |head -n 30 >>/root/.estadisticas/numeroComandosPorUsuario.txt
```

- sa-> muestra las estadísticas de ejecución de distintos programas
- -m->nos muestra por usuario
- **sort**-> lo utilizamos junto con el -k5 para ordenar tomando como referencia la quinta columna, y con el -r lo ordenamos de mayor a menor.
- **head**-> tomamos las líneas desde la "cabeza" es decir desde arriba del todo y en concreto, tomamos 30 líneas.

01111 0.0		.OIDU			
GNU nano 3.2	numero	ComandosPorUsu	iario.txt		
<u>E</u> STADÍSTICAS OBTENIDAS A DÍA Y HOR	۰۵۰				
26/04/2020	\n.				
12:51					
root	6009	3925.51re	0.60cp	0avio	2085k
pi	16	0.00re	0.00cp	Oavio	1273k
man	471	0.07re	0.00cp	Oavio	2426k
_apt	233	0.20re	0.06cp	Oavio	1309k
alvaro	94	0.20re	0.00cp	Oavio	1126k
	6823	3925.98re	0.67cp	Oavio	2067k
ESTADÍSTICAS OBTENIDAS A DÍA Y HOF	RA:				
26/04/2020					
12:52					
root	6035	3925.51re	0.60cp	0avio	2081k
		0.00re	0.00cp	0avio	1273k
man	471	0.07re	0.00ср	Oavio	2426k
_apt	233	0.20re	0.06ср	0avio	1309k
alvaro	94	0.20re	0.00ср	0avio	1126k
	6849	3925.98re	0.67cp	0avio	2064k
ESTADÍSTICAS OBTENIDAS A DÍA Y HOF	RA:				
26/04/2020					
12:53					
root	6064	3930.67re	0.60cp	Oavio	2078k
	16	0.00re	0.00cp	Oavio	1273k
man	471	0.07re	0.00cp	Oavio	2426k
_apt	233	0.20re	0.06cp	0avio	1309k
alvaro	94 6878	0.20re	0.00cp	0avio	1126k
ESTADÍSTICAS OBTENIDAS A DÍA Y HOF		3931.14re	0.67cp	0avio	2061k
ESTHUISTICHS OBTENIOHS H DIH Y HOM 26/04/2020	XH:				
2670472020 12:54					
root	6099	3930.67re	0.60cp	Oavio	2073k
oi	16	0.00re	0.80cp	Oavio	2073k 1273k
		21 líneas leíc		Vavio	12 (JK
^G Ver ayuda			J Justificar 👊	Posición	1−U Deshacer
			T Ortografía ^		
M LCCI IICII. N KCI	mpiuzui	e regai thi	- creogramia _	i i a iinca 🛚	. L Kondeel

#### estadisticas.service:

```
GNU nano 3.2 /etc/systemd/system/estadisticas.service Modificado

[Unit]

Description= Ejecutara los scripts encargados de elaborar las estadisticas

[Service]

ExecStartPre=/root/.estadisticas/tiemCadaUsuarConec.sh

ExecStartPre=/root/.estadisticas/tiemUsuarConec.sh

ExecStartPre=/root/.estadisticas/commandosMasEjecutados.sh

ExecStartPre=/root/.estadisticas/commandoPU.sh

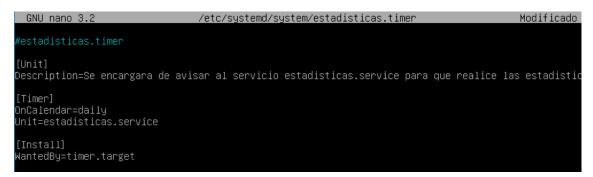
ExecStartPre=/root/.estadisticas/numeroCommandosPorUsuario.sh

[Install]

WantedBy=mult-user.target
```

Este es el fichero del servicio el cual se encargará de ejecutar todos los sh anteriormente explicados. Utilizamos "ExecStartPre" para poder ejecutar varios scripts desde el mismo servicio.

#### estadisticas.timer:



Este es el fichero encargado de ejecutar diariamente estadisticas.service, para esto utilizamos "OnCalendar=daily" (aunque para las pruebas proporcionadas para esta memoria se configuró el timer para que ejecutase el servicio una vez por minuto, con el fin de obtener resultados que poder mostrar).

# **INSTALACIÓN:**

Para realizar la instalación deberemos seguir los siguientes pasos:

- 1º Crearemos los scripts presentados al principio en mi caso se encuentran en el directorio /root/.estadisticas/ficheros.sh
- 2º Le daremos a todos los scripts permisos de ejecución a través del comando "chmod +X script.sh"
- 3º Insertaremos el fichero .service en el directorio /etc/systemd/system
- 4º Insertaremos el fichero .timer en el directorio /etc/systemd/system
- 5º Activaremos el .service y el .timer a través del comando systemctl enable fichero.service/fichero.timer.

# **VENTAJAS DE TIMER FRENTE A CRONTAB:**

Algunas de las ventajas que tienen los timer frente al cron podrían ser:

- -Los timers son servicios de systemd por lo tanto poseen las capacidades necesarias para gestionar los recursos.
- -Los timers registran sus ejecuciones en el diario de systemd por lo tanto pueden comprobarse, mientras que cron es mas costoso de verificar su correcto funcionamiento.
- -Una vez escrito un timer, activarlo y desactivarlo es mucho más sencillo, tan solo precisamos de la ejecución de una línea para hacerlo.
- -Los eventos de calendario utilizados por systemd son más precisos que cron.
- -Si se tiene en cuenta el uso de CPU, el cron la utiliza más a menudo de lo que necesita usarla un timer.