

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.

```
GNU nano 3.2                                tiemUsuarConec.sh                                Modificado
#!/bin/bash
echo "ESTADISTICAS REALIZADAS EN EL SIGUIENTE DIA Y HORA: " >> /root/.estadisticas/tiemUsuarConec.t
date +"%d/%m/%Y" >> /root/.estadisticas/tiemUsuarConec.txt
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

```
GNU nano 3.2                                tiemUsuarConec.txt
Apr 25 total      13.54
Apr 23 total      11.60
Mar 19 total      10.63
Apr 12 total       6.77
Apr  6 total       5.40
Today total       3.09
ESTADISTICAS REALIZADAS EN EL SIGUIENTE DIA Y HORA:
26/04/2020
21:54
Mar 20 total      14.47
Apr  7 total      13.84
Apr 25 total      13.54
Apr 23 total      11.60
Mar 19 total      10.63
Apr 12 total       6.77
Apr  6 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      13.54
Apr 23 total      11.60
Mar 19 total      10.63
Apr 12 total       6.77
Apr  6 total       5.40
Today total       3.12

Ver ayuda  Guardar  Buscar  Contar txt  Justificar  Posición  Deshacer
Salir      Leer fich. Reemplazar  Pegar txt  Ortografía  Ir a línea  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.

```
GNU nano 3.2          tiemCadaUsuarConec.sh          Modificado
#!/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 líneas desde la "cabeza" es decir desde arriba del todo y en concreto, tomamos 20 líneas

```
GNU nano 3.2          tiemCadaUsuarConec.txt          Modificado
total      79.24
root       72.27
alvaro     6.98
ESTADÍSTICAS REALIZADAS EN EL SIGUIENTE DIA Y HORA
26/04/2020
21:48
total      79.26
root       72.28
alvaro     6.98
ESTADÍSTICAS REALIZADAS EN EL SIGUIENTE DIA Y HORA
26/04/2020
21:49
total      79.28
root       72.30
alvaro     6.98
ESTADÍSTICAS REALIZADAS EN EL SIGUIENTE DIA Y HORA
26/04/2020
21:50
total      79.30
root       72.32
alvaro     6.98
ESTADÍSTICAS REALIZADAS EN EL SIGUIENTE DIA Y HORA
26/04/2020
21:51
total      79.32
root       72.34
alvaro     6.98

^G Ver ayuda  ^O Guardar  ^W Buscar  ^K Cortar txt  ^J Justificar  ^C Posición  M-U Deshacer
^X Salir      ^R Leer fich.  ^E Reemplazar  ^U Pegar txt  ^I Ortografía  ^_ Ir a línea  M-E Rehacer
```

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.

```
GNU nano 3.2 commandCPU.sh
#!/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.

```
GNU nano 3.2 commandCPU.txt
ESTADÍSTICAS REALIZADAS EN EL SIGUIENTE DIA Y HORA
26/04/2020
12:51
 1 158.62re 0.00cp 0avio 56256k rsyslogd
 1 7.68re 0.00cp 0avio 30528k gnuplot
17 0.00re 0.00cp 0avio 26253k (ndCPU.sh)*
 2 0.00re 0.00cp 0avio 26248k (uavio.sh)*
 1 0.00re 0.00cp 0avio 26240k (sd-sync)*
 3 0.00re 0.00cp 0avio 26240k (sario.sh)*
 1 0.00re 0.00cp 0avio 26240k (Conec.sh)*
 1 149.56re 0.00cp 0avio 25368k (sd-pam)*
 1 0.02re 0.01cp 0avio 25456k update-mime-dat
 8 0.13re 0.00cp 0avio 22696k curl
20 0.02re 0.00cp 0avio 22034k (sd-executor)*
 2 0.00re 0.00cp 0avio 11503k gtk-query-immod
 5 1.28re 0.03cp 0avio 8838k s-tui
 5 0.00re 0.00cp 0avio 8745k s-tui*
 3 0.00re 0.00cp 0avio 8449k apt-listchanges
 3 0.02re 0.01cp 0avio 6447k dpkg-preconfigu
 3 0.01re 0.00cp 0avio 6381k dpkg-preconfigu*
 3 1.42re 0.00cp 0avio 6369k journalctl
 2 0.01re 0.00cp 0avio 6292k frontend
 8 0.01re 0.00cp 0avio 6144k apt-extracttemp
10 0.18re 0.06cp 0avio 5856k http
58 0.01re 0.00cp 0avio 5834k mysql
 1 149.56re 0.00cp 0avio 5256k systemd
 1 158.62re 0.00cp 0avio 4944k systemd-logind
 3 0.01re 0.00cp 0avio 4928k py3compile
 2 0.07re 0.02cp 0avio 4902k apt
 2 0.00re 0.00cp 0avio 4874k mysqlcheck
 2 0.01re 0.00cp 0avio 4836k mysql_upgrade
 2 0.00re 0.00cp 0avio 4824k systemd-update-
[ 8844 líneas leídas ]
G Ver ayuda  O Guardar  B Buscar  K Cortar txt  J Justificar  C Posición  M-U Deshacer
X Salir  R Leer fich.  N Reemplazar  U Pegar txt  I Ortografía  A Ir a línea  W-E Rehacer
```

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.

```
GNU nano 3.2                                comandosMasEjecutados.sh                                Modificado
#!/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                                comandosMasEjecutados.txt
INFORMACION ALMACENADA A DIA Y HORA:
26/04/2020
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
```

```
GNU nano 3.2                                comandosMasEjecutados.txt
58         0.01re      0.00cp      0avio      3854k      apt-config
INFORMACION ALMACENADA A DIA Y HORA:
26/04/2020
12:52
6837      3925.98re      0.67cp      0avio      2065k
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*
443        0.01re      0.00cp      0avio      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.

```
GNU nano 3.2 numeroComandosPorUsuario.sh
#!/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.

```
GNU nano 3.2 numeroComandosPorUsuario.txt
ESTADÍSTICAS OBTENIDAS A DÍA Y HORA:
26/04/2020
12:51
root          6009    3925.51re    0.60cp    0avio    2085k
pi             16      0.00re    0.00cp    0avio    1273k
man           471     0.07re    0.00cp    0avio    2426k
lapt          233     0.20re    0.06cp    0avio    1309k
alvaro        94      0.20re    0.00cp    0avio    1126k
              6823    3925.98re    0.67cp    0avio    2067k
ESTADÍSTICAS OBTENIDAS A DÍA Y HORA:
26/04/2020
12:52
root          6035    3925.51re    0.60cp    0avio    2081k
pi             16      0.00re    0.00cp    0avio    1273k
man           471     0.07re    0.00cp    0avio    2426k
lapt          233     0.20re    0.06cp    0avio    1309k
alvaro        94      0.20re    0.00cp    0avio    1126k
              6849    3925.98re    0.67cp    0avio    2064k
ESTADÍSTICAS OBTENIDAS A DÍA Y HORA:
26/04/2020
12:53
root          6064    3930.67re    0.60cp    0avio    2078k
pi             16      0.00re    0.00cp    0avio    1273k
man           471     0.07re    0.00cp    0avio    2426k
lapt          233     0.20re    0.06cp    0avio    1309k
alvaro        94      0.20re    0.00cp    0avio    1126k
              6878    3931.14re    0.67cp    0avio    2061k
ESTADÍSTICAS OBTENIDAS A DÍA Y HORA:
26/04/2020
12:54
root          6099    3930.67re    0.60cp    0avio    2073k
pi             16      0.00re    0.00cp    0avio    1273k
[ 2421 líneas leídas ]
Ver ayuda  Guardar  Buscar  Cortar txt  Justificar  Posición  M-U  Deshacer
Salir  Leer fich.  Reemplazar  Pegar txt  Ortografía  Ir a línea  M-E  Rehacer
```

estadisticas.service:

```
GNU nano 3.2 /etc/systemd/system/estadisticas.service Modificado
[Unit]
Description= Ejecutara los scripts encargados de elaborar las estadísticas
[Service]
ExecStartPre=/root/.estadisticas/tiemCadaUsuarConec.sh
ExecStartPre=/root/.estadisticas/tiemUsuarConec.sh
ExecStartPre=/root/.estadisticas/comandosMasEjecutados.sh
ExecStartPre=/root/.estadisticas/commandCPU.sh
ExecStart=/root/.estadisticas/numeroComandosPorUsuario.sh
[Install]
WantedBy=multi-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:

```
GNU nano 3.2 /etc/systemd/system/estadisticas.timer Modificado
#estadisticas.timer
[Unit]
Description=Se encargara de avisar al servicio estadisticas.service para que realice las estadísticas
[Timer]
OnCalendar=daily
Unit=estadisticas.service
[Install]
WantedBy=timer.target
```

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.