

Para poder acceder a la información de cada proceso, primero hay que ver que PID tiene cada proceso y cuales están activos, para ello uso el comando “ps aux”:

```

root      481  0.0  0.0      0  0 ?      S<  16:29  0:00 [ext4-rsv-conver]
root      487  0.0  0.0      0  0 ?      S<  16:29  0:00 [xfsalloc]
root      488  0.0  0.0      0  0 ?      S<  16:29  0:00 [xfs_mru_cache]
root      489  0.0  0.0      0  0 ?      S<  16:29  0:00 [xfs_buf/dm-2]
root      490  0.0  0.0      0  0 ?      S<  16:29  0:00 [xfs-data/dm-2]
root      491  0.0  0.0      0  0 ?      S<  16:29  0:00 [xfs-conu/dm-2]
root      492  0.0  0.0      0  0 ?      S<  16:29  0:00 [xfs-cil/dm-2]
root      493  0.0  0.0      0  0 ?      S  16:29  0:00 [xfsaild/dm-2]
root      497  0.0  0.0      0  0 ?      S<  16:29  0:00 [ext4-rsv-conver]
message+  515  0.0  0.1  39120 2088 ?    Ss  16:29  0:00 dbus-daemon --system --fork
537       537  0.0  0.1  43456 3256 ?    Ss  16:29  0:00 /lib/systemd/systemd-logind
syslog    539  0.0  0.1 255848 2824 ?    Ssl 16:29  0:00 rsyslogd
548       548  0.0  0.0 15544 1720 ?    S  16:29  0:00 upstart-file-bridge --daemon
root      604  0.0  0.0      0  0 ?      S<  16:29  0:00 [ltn_swap]
root      798  0.0  0.0 15264 236 ?      S  16:29  0:00 upstart-socket-bridge --daemon
root      903  0.0  0.1 10236 3256 ?    Ss  16:29  0:00 dhclient -t -c -pf /run/dhclient.e
root      1070  0.0  0.1 18336 2160 tty4  Ss+ 16:29  0:00 /sbin/getty -8 38400 tty4
root      1073  0.0  0.1 18332 2052 tty5  Ss+ 16:29  0:00 /sbin/getty -8 38400 tty5
root      1078  0.0  0.1 18336 2164 tty2  Ss+ 16:29  0:00 /sbin/getty -8 38400 tty2
root      1079  0.0  0.1 18336 2056 tty3  Ss+ 16:29  0:00 /sbin/getty -8 38400 tty3
root      1082  0.0  0.1 18336 2160 tty6  Ss+ 16:29  0:00 /sbin/getty -8 38400 tty6
root      1108  0.0  0.2 61372 5340 ?    Ss  16:29  0:00 /usr/sbin/sshd -D
root      1130  0.0  0.1 23660 2300 ?    Ss  16:29  0:00 cron
daemon    1132  0.0  0.0 19144 160 ?      Ss  16:29  0:00 atd
root      1176  0.0  0.0 4372 1576 ?    Ss  16:29  0:00 acpid -c /etc/acpi/events -s /var/
mysql     1205  0.1  2.7 558604 56140 ?   Ssl 16:29  0:00 /usr/sbin/mysqld
root      1222  0.0  1.0 276380 22140 ?   Ss  16:29  0:00 /usr/sbin/apache2 -k start
www-data  1236  0.0  0.3 276404 7404 ?    S  16:29  0:00 /usr/sbin/apache2 -k start
www-data  1237  0.0  0.3 276404 7404 ?    S  16:29  0:00 /usr/sbin/apache2 -k start
www-data  1238  0.0  0.3 276404 7404 ?    S  16:29  0:00 /usr/sbin/apache2 -k start
www-data  1239  0.0  0.3 276404 7404 ?    S  16:29  0:00 /usr/sbin/apache2 -k start
www-data  1240  0.0  0.3 276404 7404 ?    S  16:29  0:00 /usr/sbin/apache2 -k start
root      1278  0.0  0.1 78188 3568 tty1  Ss  16:29  0:00 /bin/login --
root      1386  0.0  0.0      0  0 ?      S  16:30  0:00 [kauditd]
jose lu   1512  0.0  0.2 25300 5648 tty1  S  16:30  0:00 -bash
jose lu   1554  0.0  0.1 21024 2588 tty1  R+  16:42  0:00 ps aux
jose lu@ubuntu:/proc$ _

```

Al escribir `/proc/[pid]`, obtenemos la información del proceso con la `pid` indicada:

```

joseph@ubuntu:/$ cd /proc/
joseph@ubuntu:/proc$ cd 1512
joseph@ubuntu:/proc/1512$ ls
attr          comm          fd             map_files     net           pagemap       sessionid     status        sysctl
autogroup     coredump_filter fdinfo         maps          ns            personality   setgroups     syscall
auxv          cpuset        gid_map        mem           numa_maps     projid_map     smaps         task
cgroup        cwd           io             mountinfo     oom_adj       root          stack         timers
clear_refs    environ       limits         mounts         oom_score     sched          stat          uid_map
cmdline       exe           loginuid       mountstats    oom_score_adj schedstat      statn         uchan
joseph@ubuntu:/proc/1512$ _

```

Para poder acceder a parámetros del kernel escribimos /proc/sys y obtenemos, aquí podemos modificar parámetros del propio kernel:

```
joseph@ubuntu:/$ cd /proc/sys
joseph@ubuntu:/proc/sys$ ls
abi debug dev fs kernel net vm
joseph@ubuntu:/proc/sys$ ls
abi debug dev fs kernel net vm
joseph@ubuntu:/proc/sys$
```

CentOS:

Al igual que en Ubuntu, en CentOS podemos acceder a estos directorios que contienen la misma información que en Ubuntu.

En el directorio /proc encontramos:

[illegible]

Si accedemos por ejemplo a meminfo obtenemos informacion variada de la memoria de nuestro sistema:

```

joselu@localhost:~$ cat /proc/meminfo
MemTotal: 1026440 kB
MemFree: 387796 kB
MemAvailable: 1185326 kB
Buffers: 798 kB
Cached: 384596 kB
SwapCached: 0 kB
Active: 486888 kB
Inactive: 191772 kB
Active(anon): 427594 kB
Inactive(anon): 8490 kB
Active(file): 62384 kB
Inactive(file): 311872 kB
Unevictable: 0 kB
Mlocked: 0 kB
SwapTotal: 1391972 kB
SwapFree: 1348372 kB
Dirty: 12 kB
Writeback: 0 kB
CommitTotal: 688796 kB
CommitFree: 57440 kB
CommitReserve: 6126 kB
CommitLimit: 75796 kB
PhysicalMem: 26896 kB
SwpTotal: 45896 kB
KernelStack: 16500 kB
PageTables: 26252 kB
NFS_Unstable: 0 kB
AnonHugePages: 0 kB
WritebackTmp: 0 kB
DMAFree: 14600 kB
Committed_AS: 2924754 kB
VmallocTotal: 3435973836 kB
VmallocUsed: 285132 kB
```

Igual ocurre al acceder a la información de los procesos y del kernel de CentOS ya que tanto Ubuntu como CentOS son sistemas basados en Linux.

Cuestión 2:

Para monitorizar el sistema de archivos es muy útil el comando df el cual nos indica el espacio total, ocupado y libre de cada uno de los sistemas de ficheros del sistema además de indicar donde están montados:

```

Ubuntu 14.04.3 LTS ubuntu tty1
ubuntu login: joselu
Password:
Last login: Fri Dec 11 12:17:32 CET 2015 on tty1
Welcome to Ubuntu 14.04.3 LTS (GNU/Linux 3.19.0-25-generic x86_64)

 * Documentation:  https://help.ubuntu.com/

System information as of Fri Dec 11 12:39:58 CET 2015

System load:  0.15           Memory usage: 4%        Processes:   191
Usage of /home: 0.2% of 921MB  Swap usage:  0%        Users logged in: 0

Graph this data and manage this system at:
https://landscape.canonical.com/

67 packages can be updated.
30 updates are security updates.

joselu@ubuntu:~$ df
S.ficheros                blocks de 1K  Usados  Disponibles  Usos%  Montado en
/dev/mapper/grupoCESI-root 4674688 1360988  3053196    31% /
none                        4         0         4         0% /sys/fs/cgroup
udev                      1095056   4    1095052    1% /dev
tmpfs                     203240    980    202260    1% /run
none                       5120     0     5120     0% /run/lock
none                      1016192   0    1016192    0% /run/shm
none                       102400    0     102400    0% /run/user
/dev/mapper/grupoCESI-home 943128    1244    939560    1% /home
/dev/mapper/grupoCESI-homeEXT2 959512    1204    958308    1% /homeEXT2
/dev/mapper/grupoCESI-homeEXT3 943128    1224    939164    1% /homeEXT3
/dev/mapper/grupoCESI-homeXFS 970048    32928   937120    4% /homeXFS
joselu@ubuntu:~$
```

El comando df ofrece más información, para ello hay que usar ciertos parámetros que podemos ver tecleando “man df” y saliendo así el manual del comando df. Uno de esos parámetros es el -h que hace que las unidades sean más fácil de visualizar

```
-H, --si
    likewise, but use powers of 1000 not 1024

-i, --inodes
    list inode information instead of block usage

-k      like --block-size=1K

-l, --local
    limit listing to local file systems

--no-sync
    do not invoke sync before getting usage info (default)

--output[=FIELD_LIST]
    use the output format defined by FIELD_LIST, or print all fields if FIELD_LIST is
    omitted.

-P, --portability
    use the POSIX output format

joselu@ubuntu:~$ df -h
df: <=>: No existe el archivo o el directorio
df: <=>: No existe el archivo o el directorio
joselu@ubuntu:~$ df -h
$ ficheros
Tamaño Usados  Disp Usos% Montado en
/dev/mapper/grupoCESI-root
none 4,5G 1,3G 3,0G 31% /
udev 4,0K 0 4,0K 0% /sys/fs/cgroup
udev 982M 4,0K 982M 1% /dev
tmpfs 199M 980K 198M 1% /run
none 5,0M 0 5,0M 0% /run/lock
none 993M 0 993M 0% /run/shm
none 100M 0 100M 0% /run/user
/dev/mapper/grupoCESI-home 922M 1,3M 857M 1% /home
/dev/mapper/grupoCESI-homeEXT2 938M 1,2M 889M 1% /home/EXT2
/dev/mapper/grupoCESI-homeEXT3 922M 1,2M 873M 1% /home/EXT3
/dev/mapper/grupoCESI-homeXFS 948M 33M 916M 4% /home/XFS
joselu@ubuntu:~$
```

Para la monitorización de la memoria del sistema existe el comando free el cual nos da la información sobre la memoria física del sistema, total, libre, compartida, etc. Usando los parámetros como -t para ver la memoria total o -b para ver la cantidad en bytes, obtenemos un análisis más minuciosa de la memoria:

```
-l, --lohi
    Show detailed low and high memory statistics.

-o, --old
    Display the output in old format, the only difference being this option will dis-
    able the display of the "buffer adjusted" line.

-s, --seconds seconds
    Continuously display the result delay seconds apart. You may actually specify any
    floating point number for delay, usleep(3) is used for microsecond resolution delay
    times.

--si Use power of 1000 not 1024.

-t, --total
    Display a line showing the column totals.

--help Print help.

-V, --version
    Display version information.

FILES
/proc/meminfo
joselu@ubuntu:~$ free -t
total          usado         libre      compart.    buffers    alnac.
Mem: 2032384 289168 1743216 6396 18100 117052
-/+ buffers/cache: 154916 1878368
Intercambio: 974044 0 974044
Total: 3007228 289168 2718060
joselu@ubuntu:~$ free -b
total          usado         libre      compart.    buffers    alnac.
Mem: 2091161216 296108032 1795053184 6549504 18534400 119861248
-/+ buffers/cache: 157712304 1937340880
Intercambio: 998240256 0 998240256
joselu@ubuntu:~$ _
```

Para los procesos, un comando muy útil es top que nos muestra en tiempo real la información más relevante de cada proceso ordenándolos por uso de %CPU. Tecleando nada más que top nos aprecian todos los procesos del sistema y su información, pero un parámetro muy útil es "-p" ya que con el se puede especificar el PID del proceso que quiere observar para así poder centrarte en ese, en este caso busco la PID del propio top:

```
top 12:55:14 up 15 min, 1 user, load average: 0.00, 0.01, 0.05
Tasks: 1 total, 0 sleeping, 0 idle, 1 detector, 0 zombie
PID: 0.0 usrio, 0.0 nice, 0.0 shared, 0.0 hard, 0.0 in cache, 0.0 hard int, 0.0 s
118 Mem: 289168 total, 296108 used, 179505 free, 18534 buffers
TID Swap: 974044 total, 0 used, 974044 free, 117320 cached Mem

PID CPU% MEM% VSZ% RSS% S% B% D% T% C%
118 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0
118 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0
```

Para tener un control de los usuarios, podemos usar el comando `who` el cual nos muestra los usuarios conectados además de una serie de datos. Un parámetro muy útil es “-a” el cual nos indica toda la información sobre el comando `who`, como cuando se realizó el ultimo arranque del sistema:

```
-a, --all
    same as -b -d --login -p -r -t -T -u

-b, --boot
    time of last system boot

-d, --dead
    print dead processes

-H, --heading
    print line of column headings

--ips
    print ips instead of hostnames, with --lookup, canonicalizes based on stored IP, if
    available, rather than stored hostname

-l, --login
    print system login processes

--lookup
    attempt to canonicalize hostnames via DNS

-n
    only hostname and user associated with stdin

-p, --process
    print active processes spawned by init
joselu@ubuntu:~$ who -a
arranque del sistema 2015-12-11 12:40
run-level 2 2015-12-11 12:40
LOGIN tty1 2015-12-11 12:40 1004 id=4
LOGIN tty5 2015-12-11 12:40 1007 id=5
LOGIN tty2 2015-12-11 12:40 1012 id=2
LOGIN tty3 2015-12-11 12:40 1013 id=3
LOGIN tty6 2015-12-11 12:40 1016 id=6
joselu - tty1 2015-12-11 12:40 1445
joselu@ubuntu:~$ who -b
arranque del sistema 2015-12-11 12:40
joselu@ubuntu:~$
```

Cuestión 3:

Para guardar la información del comando `vmstat` cada 5 segundos durante 5 min solo hay que escribir por el terminal:

`vmstat 5 300 >> vmstat.res`

El primer parámetro nos indica cada cuantos segundos se tomara la medición, el segundo durante cuánto tiempo. El operador `>>` redirige el flujo de datos al fichero `vmstat.res`

Cuestión 4:

Para sacar las distintas opciones que dispone el comando `sar`, usare “man sar” (especifico varios de ellos ya que son muchos):

- -A: Muestra todas las posibles actividades.
- -B: Reporta el comportamiento del fichero de paginación.
- -b: Estadísticas de transferencia de entrada y salida.
- -C: Cuando se lee datos de un fichero, muetsra los comentarios que han sido insertados por sadc.
- -d: Reporta la actividad de cada dispositivo.

- -e [hh:mm:ss]: Define el tiempo de final del estudio.
- -f [filename]: extrae datos de un fichero filename.
- -H: Reporta estadísticas de utilización de paginas grandes.
- -h: Muestra un pequeño mensaje de ayuda.
- -i interval: Selecciona los datos grabados en el intervalo dado en segundos por interval.
- -u: Comprueba el uso de la CPU.
- -r: Comprueba la memoria no utilizada.
- -y: Comprueba la actividad del terminal.
- ...

Cuestión 5:

En cuanto a la utilización del procesador uso el comadno “sar –u 2 4”:

```
Linux 3.10.0-229.el7.x86_64 (localhost.localdomain) 04/12/15 _x86_64_
(1 CPU)

11:29:37      CPU      %user      %nice      %system      %iowait      %steal      %idle
11:29:39      all       16,24        0,00        4,06        0,00        0,00       79,70
11:29:41      all       26,32        0,00        9,47        0,00        0,00       64,21
11:29:43      all       16,67        0,00        2,02        0,00        0,00       81,31
11:29:45      all       14,57        0,00        2,51        0,00        0,00       82,91
Media:        all       18,37        0,00        4,46        0,00        0,00       77,17
```

En cuanto a la carga media del sistema uso “sar –A”

```
Linux 3.10.0-229.el7.x86_64 (localhost.localdomain) 05/12/15 _x86_64_
(1 CPU)

13:18:22      LINUX RESTART

13:28:01      CPU      %usr      %nice      %sys      %iowait      %steal      %irq
13:38:02      %soft  %guest  %gnice  %idle      2,31      1,04      0,00      0,00
13:38:02      all      10,78      0,01      85,72
13:38:02      0,14      0,00      0,00      85,72
13:38:02      0      10,78      0,01      2,31      1,04      0,00      0,00
13:38:02      0,14      0,00      0,00      85,72

Media:        CPU      %usr      %nice      %sys      %iowait      %steal      %irq
Media:        %soft  %guest  %gnice  %idle      2,31      1,04      0,00      0,00
Media:        0,14      0,00      0,00      85,72
Media:        0      10,78      0,01      2,31      1,04      0,00      0,00
Media:        0,14      0,00      0,00      85,72

13:28:01      proc/s   cswch/s
13:38:02      1,14     144,11
Media:        1,14     144,11

13:28:01      pswpin/s pswpout/s
13:38:02      0,00     0,00
Media:        0,00     0,00

13:28:01      pgpgin/s pgpgout/s   fault/s   majflt/s   pgfree/s pgscank/s pgscand/
s pgsteal/s   %vmeff
```

En cuanto a la paginación uso “sar –B”:

```
Linux 3.10.0-229.el7.x86_64 (localhost.localdomain) 05/12/15 _x86_64_
(1 CPU)

13:18:22          LINUX RESTART

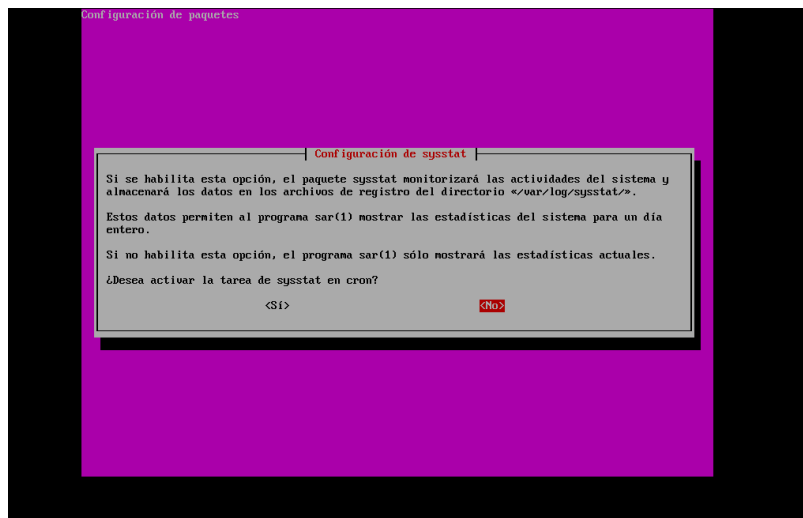
13:20:01      pgpgin/s pgpgout/s   fault/s   majflt/s   pgfree/s pgscank/s pgscand/
s pgsteal/s      %vmeff
13:30:02      223,69      11,04      660,31      0,45      999,65      0,00      0,0
0      0,00      0,00
Media:      223,69      11,04      660,31      0,45      999,65      0,00      0,0
0      0,00      0,00
```

Cuestión 6:

Para instalar el paquete sysstat en Ubuntu debemos hacer:

1. Sudo apt-get install sysstat
2. Service sysstat start
3. Sudo dpkg-reconfigure sysstat

Una vez realizado esto, nos saldrá la ventana para dar permiso a sysstat en cron:



Ahora iremos al archivo `/etc/crontab` y lo modificaremos para que realice la medición cada 10 min:

```
GNU nano 2.2.6                                Archivo: 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
PATH=/usr/local/sbin:/usr/local/bin:/sbin:/bin:/usr/sbin:/usr/bin

# m h dom mon dow user  command
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 )
#
```

Cuestión 7:

Para ejecutar sar en modo interactivo durante 5 minutos con una frecuencia de 30 segundos hay que teclear:

```
sar 30 10
```

Donde 30 sería los segundos que tardaría en refrescarse el estudio y 10 las iteraciones de 30 segundos que en total harían 5 minutos.

```
Media:          all          0,00      0,00      0,11      0,01      0,00      99,88
joselu@ubuntu:~$ sar 30 10
Linux 3.19.0-25-generic (ubuntu)          11/12/15          _x86_64_          (1 CPU)

16:50:13      CPU      %user      %nice      %system      %iowait      %steal      %idle
16:50:43      all      0,00      0,00      0,07      0,00      0,00      99,93
16:51:13      all      0,03      0,00      0,07      0,00      0,00      99,90
16:51:43      all      0,00      0,00      0,10      0,00      0,00      99,90
16:52:13      all      0,00      0,00      0,10      0,00      0,00      99,90
16:52:43      all      0,00      0,00      0,10      0,00      0,00      99,90
16:53:13      all      0,00      0,00      0,10      0,00      0,00      99,90
16:53:43      all      0,00      0,00      0,10      0,00      0,00      99,90
16:54:13      all      0,00      0,00      0,13      0,00      0,00      99,87
16:54:43      all      0,00      0,00      0,13      0,00      0,00      99,87
16:55:13      all      0,03      0,00      0,17      0,07      0,00      99,73
Media:          all          0,01      0,00      0,11      0,01      0,00      99,88
joselu@ubuntu:~$
```

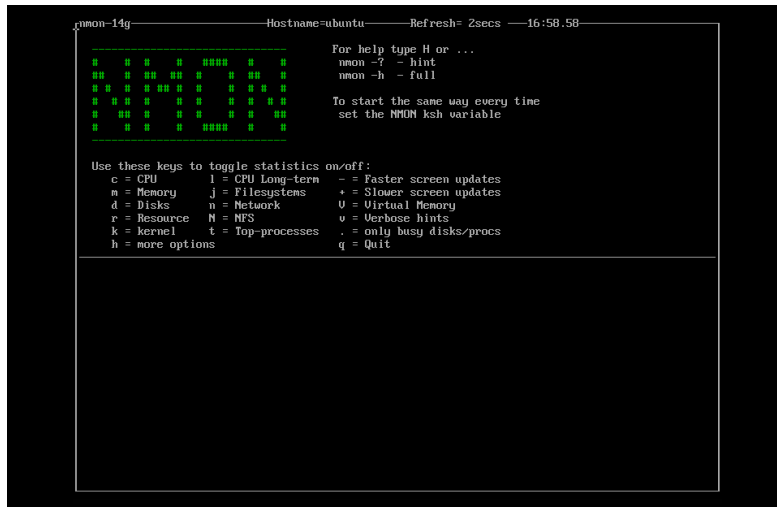
Como no he ejecutado ninguna carga a la CPU, los datos reflejan que la CPU estaba al 0,00 en la mayoría de los instantes.

Cuestión 8:

NMON:

Nos permite personalizar más la información que nos muestra, cuando lo ejecutamos nos muestra varias opciones, nos brinda la información de forma

modular, o sea, si queremos que solo nos muestre la informacion relacionada con la CPU presionamos la tecla de la opción que deseamos:

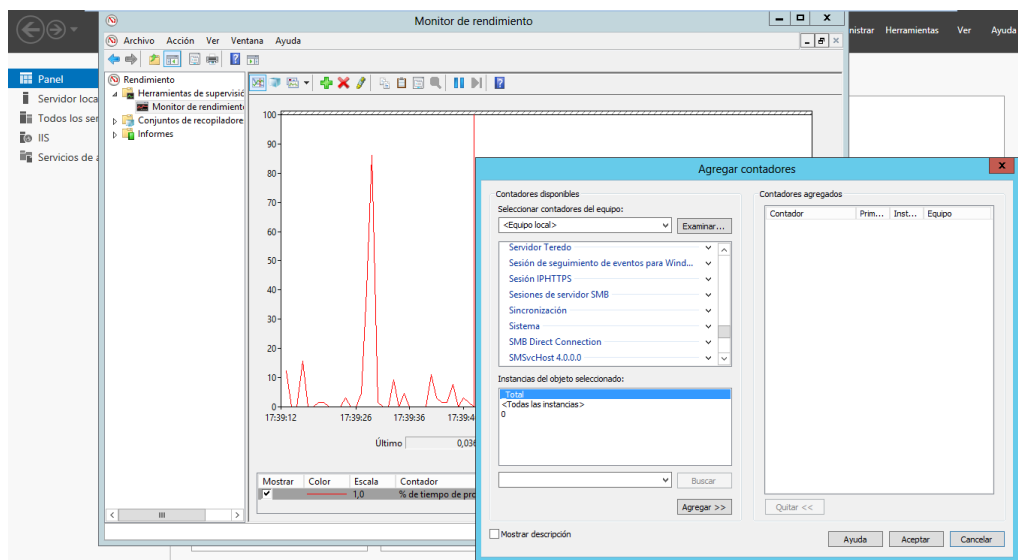


tcpdump:

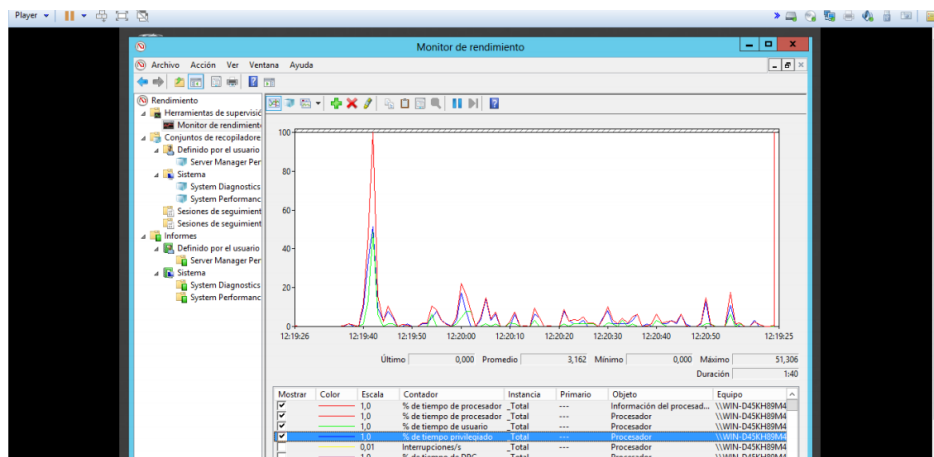
Es un analizador de paquetes de conexión. Usando tcpdump puedes capturar paquetes y analizarlos para poder así detectar cuellos de botella potenciales.

Cuestión 9:

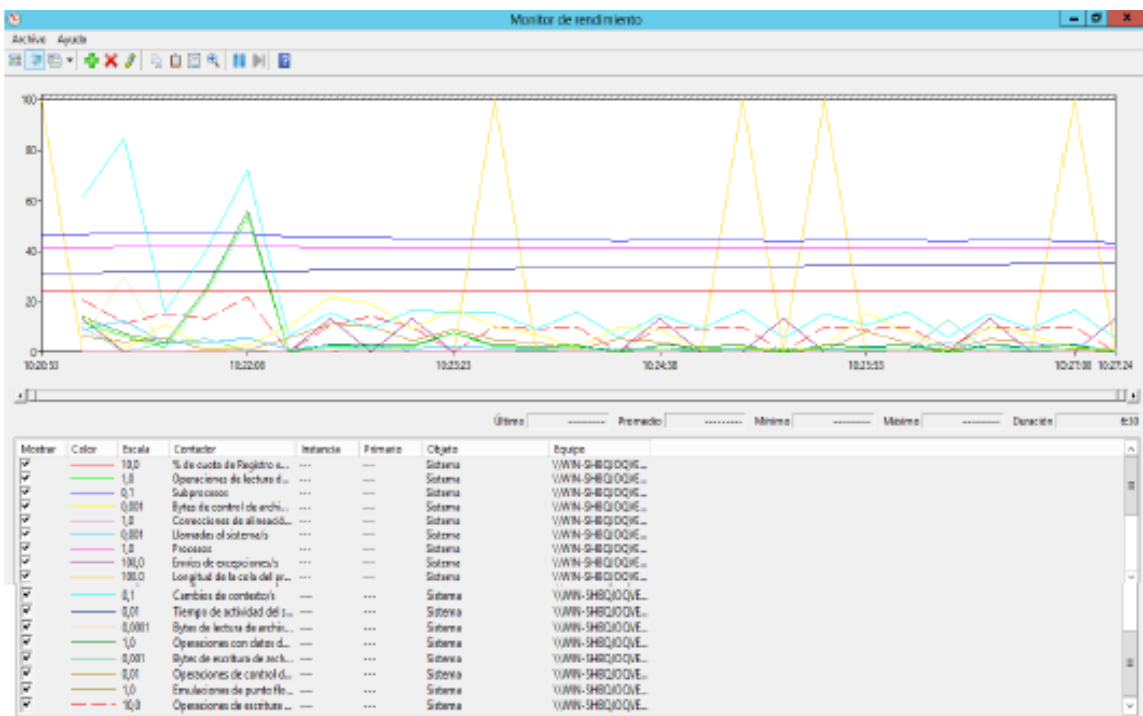
Para acceder a esta opción hay que ir al administrador del servidor y al monitor de rendimiento:



Una vez dentro hay que señalar lo que se quiere supervisar, dando lugar a una gráfica de esta manera:



Cuestión 10:



Cuestión 11:

Crear nuevo conjunto de recopiladores de datos.

¿Cómo desea crear este nuevo conjunto de recopiladores de datos?

Nombre:
Ejercicio_11

☒ Crear a partir de una plantilla (recomendado)
[¿Cómo se trabaja con plantillas?](#)

☐ Crear manualmente (avanzado)
[¿Cómo se eligen manualmente los recopiladores de datos?](#)

Siguiente Finalizar Cancelar

Crear nuevo conjunto de recopiladores de datos.

¿Qué tipo de datos desea incluir?

☒ Crear registros de datos

- ☒ Contador de rendimiento
- ☐ Datos de seguimiento de eventos
- ☐ Información de configuración del sistema

☐ Alerta del contador de rendimiento

Siguiente Finalizar Cancelar

Contadores disponibles

Seleccionar contadores del equipo:
\\WIN-7QSAVC59C30 Examinar...

ServiceModelOperation 4.0.0.0
ServiceModelService 4.0.0.0
Servicio de estado ASP.NET
Servicio FTP de Microsoft
Servicio HTTP
Servicio web
Bytes de ancho de banda bloqueados actualmente.
Bytes enviados por segundo

Instancias del objeto seleccionado:
Total
Todos los instancias>
Default Web Site

Buscar Agregar >>

Contadores agregados

Contador	Prim...	Inst...	Equipo
Procesador			
*	---	*	\\WIN-7QSAV...
Proceso			
*	---	*	\\WIN-7QSAV...
Servicio web			
*	---	*	\\WIN-7QSAV...

Quitar <<

Ayuda Aceptar Cancelar

Crear nuevo conjunto de recopiladores de datos.

¿Qué contadores de rendimiento desea registrar?

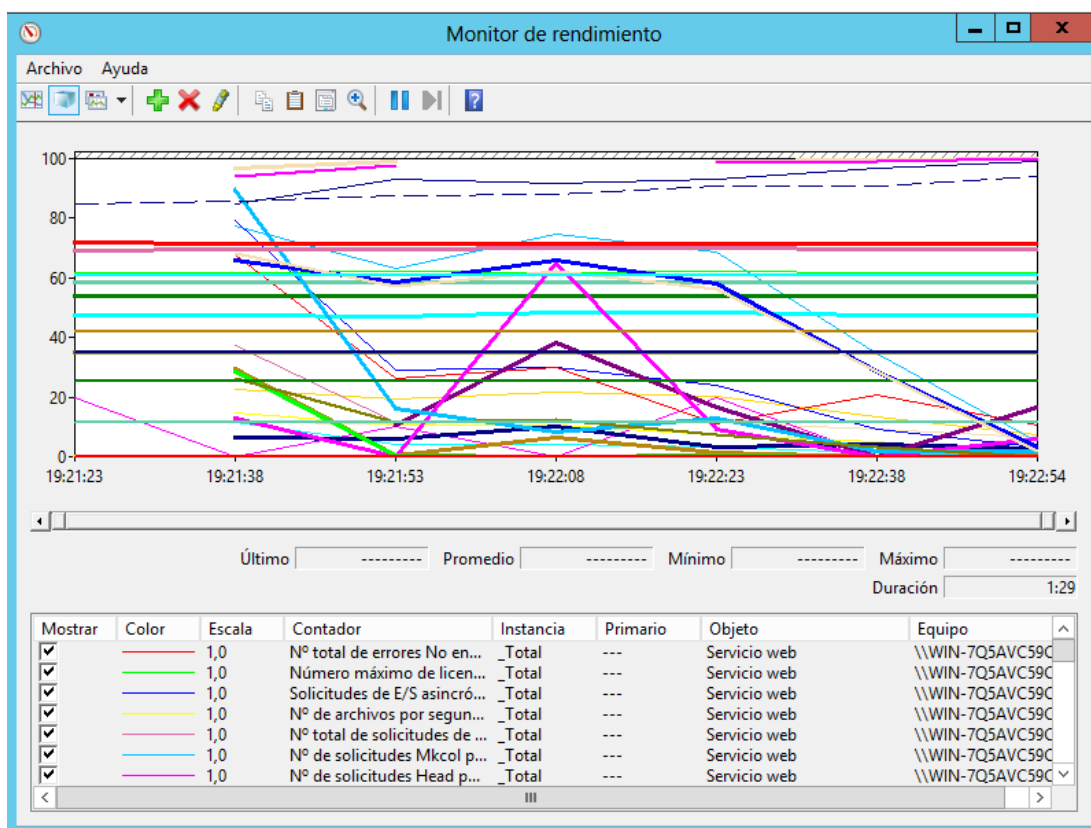
Contadores de rendimiento:

\\WIN-7Q5AVC59C3O\\Procesador(*)
 \\WIN-7Q5AVC59C3O\\Proceso(*)
 \\WIN-7Q5AVC59C3O\\Servicio web(*)

Agregar...
 Quitar

Intervalo de muestra: 15 Unidades: Segundos

Siguiente Finalizar Cancelar



Cuestión 12:

AIDA64:

Encargada de realizar un extenso y detallado análisis de sistema. Incluye módulos de identificación del hardware de la computadora, módulos de diagnóstico, un conjunto bastante completo de pruebas de rendimiento o benchmarks. Toda la información del sistema, además de mostrarla convenientemente ordenada en la pantalla, permite crear completos informes con dichos datos.

Everest:

Es un sistema de diagnóstico orientado a los usuarios domésticos. Ofrece un sistema de información y diagnóstico muy fiable, además de incluir test de memoria, monitorización de hardware, etc. Ofrece la posibilidad de crear informes en formato HTML y TXT.

Preguntas Breves

Primera pregunta:

¿Cuál es la principal diferencia entre los comandos “ps” y “top” en Linux?

“ps” muestra la información de los procesos que se están ejecutando en ese instante, mientras que “top” actualiza constantemente esta información añadiendo además más información referente a los procesos.

Segunda pregunta:

¿Qué se encuentra en el fichero “crontab”?

Es un archivo de texto que guarda una lista de comandos a ejecutar en un tiempo especificado por el usuario. Crontab verifica la fecha y hora en la que se debe ejecutar el script o el comando escrito en él y lo ejecuta.

FUENTES:

<http://www.thegeekstuff.com/2011/12/linux-performance-monitoring-tools/>

<https://es.wikipedia.org/wiki/>

<http://www.dsi.uclm.es/personal/AntonioBueno/ESI/monitor%20en%20linux.pdf>