

INGENIERÍA DE SERVIDORES (2016-2017)
GRADO EN INGENIERÍA INFORMÁTICA
UNIVERSIDAD DE GRANADA

Memoria Práctica 3

Sergio Samaniego Martínez

8 de diciembre de 2016

Índice

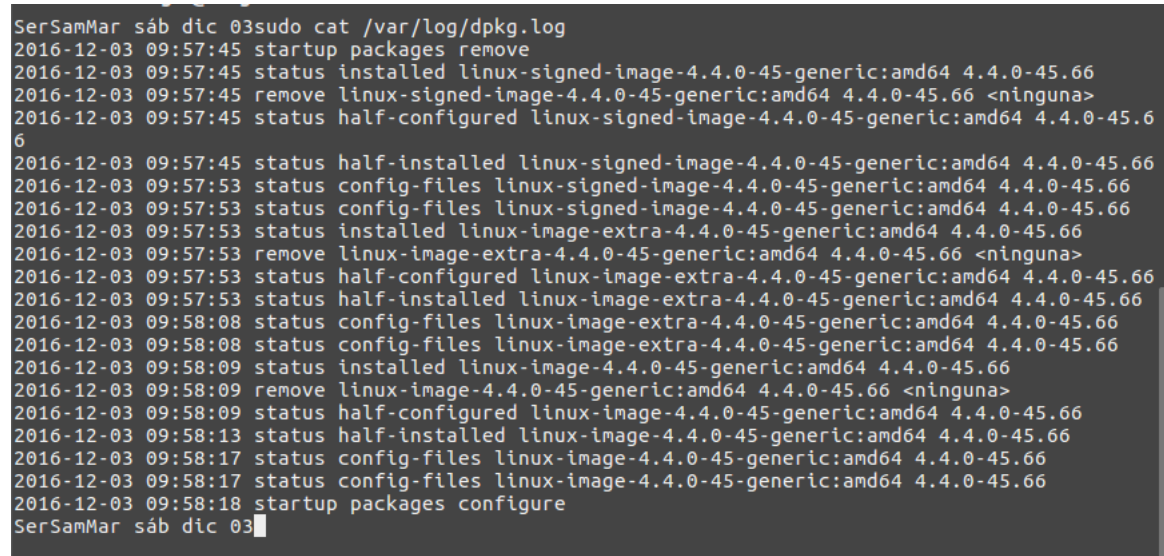
1. Cuestión 1	4
1.1. ¿Qué archivo le permite ver qué programas se han instalado con el gestor de paquetes?	4
1.2. ¿Qué significan las terminaciones .1.gz o .2.gz de los archivos en ese directorio?	4
2. Cuestión 2	5
2.1. ¿qué archivo ha de modificar para programar una tarea? Escriba la línea necesaria para ejecutar una vez al día una copia del directorio /codigo a /seguridad/ \$fecha donde \$fecha es la fecha actual (puede usar el comando date). . . .	5
3. Cuestión 3	6
3.1. Pruebe a ejecutar el comando, conectar un dispositivo USB y vuelva a ejecutar el comando. Copie y pegue la salida del comando. (considere usar dmesg tail). Comente qué observa en la información mostrada.	6
4. Cuestión 4	7
4.1. Ejecute el monitor de “System Performance” y muestre el resultado. Incluya capturas de pantalla comentando la información que aparece.	7
5. Cuestión 5	9
5.1. Cree un recopilador de datos definido por el usuario (modo avanzado) que incluya tanto el contador de rendimiento como los datos de seguimiento: Todos los referentes al procesador, al proceso y al servicio web. Intervalo de muestra 15 segundos. Almacene el resultado en el directorio Escritorio/logs. Incluya las capturas de pantalla de cada paso.	10
6. Cuestión 6	18
6.1. Visite la web del proyecto y acceda a la demo que proporcionan (http://demo.munin-monitoring.org/) donde se muestra cómo monitorizan un servidor. Monitoree varios parámetros y haga capturas de pantalla de lo que está mostrando comentando qué observa.	18
7. Cuestión 7	21
7.1. Escriba un breve resumen sobre alguno de los artículos donde se muestra el uso de strace o busque otro y coméntelo. .	21

8. Cuestión 8	21
8.1. Escriba un script en Python o PHP y analice su comportamiento usando el profiler presentado.	21
9. Cuestión 9	23
9.1. Acceda a la consola mysql (o a través de phpMyAdmin) y muestre el resultado de mostrar el "profile" de una consulta (la creación de la BD y la consulta la puede hacer libremente).	23

1. Cuestión 1

1.1. ¿Qué archivo le permite ver qué programas se han instalado con el gestor de paquetes?

El archivo que nos permite ver qué programas se han instalado con el gestor de paquetes es el archivo de registro de dpkg, el cual se encuentra en /var/log. El archivo de registro es dpkg.log y tendrá las últimas acciones hechas con el gestor de paquetes. [6]



```
SerSamMar sáb dic 03 sudo cat /var/log/dpkg.log
2016-12-03 09:57:45 startup packages remove
2016-12-03 09:57:45 status installed linux-signed-image-4.4.0-45-generic:amd64 4.4.0-45.66
2016-12-03 09:57:45 remove linux-signed-image-4.4.0-45-generic:amd64 4.4.0-45.66 <ninguna>
2016-12-03 09:57:45 status half-configured linux-signed-image-4.4.0-45-generic:amd64 4.4.0-45.66
2016-12-03 09:57:45 status half-installed linux-signed-image-4.4.0-45-generic:amd64 4.4.0-45.66
2016-12-03 09:57:53 status config-files linux-signed-image-4.4.0-45-generic:amd64 4.4.0-45.66
2016-12-03 09:57:53 status config-files linux-signed-image-4.4.0-45-generic:amd64 4.4.0-45.66
2016-12-03 09:57:53 status installed linux-image-extra-4.4.0-45-generic:amd64 4.4.0-45.66
2016-12-03 09:57:53 remove linux-image-extra-4.4.0-45-generic:amd64 4.4.0-45.66 <ninguna>
2016-12-03 09:57:53 status half-configured linux-image-extra-4.4.0-45-generic:amd64 4.4.0-45.66
2016-12-03 09:57:53 status half-installed linux-image-extra-4.4.0-45-generic:amd64 4.4.0-45.66
2016-12-03 09:58:08 status config-files linux-image-extra-4.4.0-45-generic:amd64 4.4.0-45.66
2016-12-03 09:58:08 status config-files linux-image-extra-4.4.0-45-generic:amd64 4.4.0-45.66
2016-12-03 09:58:09 status installed linux-image-4.4.0-45-generic:amd64 4.4.0-45.66
2016-12-03 09:58:09 remove linux-image-4.4.0-45-generic:amd64 4.4.0-45.66 <ninguna>
2016-12-03 09:58:09 status half-configured linux-image-4.4.0-45-generic:amd64 4.4.0-45.66
2016-12-03 09:58:13 status half-installed linux-image-4.4.0-45-generic:amd64 4.4.0-45.66
2016-12-03 09:58:17 status config-files linux-image-4.4.0-45-generic:amd64 4.4.0-45.66
2016-12-03 09:58:17 status config-files linux-image-4.4.0-45-generic:amd64 4.4.0-45.66
2016-12-03 09:58:18 startup packages configure
SerSamMar sáb dic 03
```

Figura 1.1: Muestra del archivo dpkg.log

1.2. ¿Qué significan las terminaciones .1.gz o .2.gz de los archivos en ese directorio?

Las terminaciones de .1.gz o .2.gz son anteriores archivos de registro los cuales han sido comprimidos por el demonio cron.daily. Este hace una rotación que se encarga de comprimir los archivos añadiéndoles la extensión .1.gz, .2.gz, etc volviendo a crear un nuevo archivo vacío. [9]

2. Cuestión 2

- 2.1. ¿qué archivo ha de modificar para programar una tarea? Escriba la línea necesaria para ejecutar una vez al día una copia del directorio /codigo a /seguridad/\$fecha donde \$fecha es la fecha actual (puede usar el comando date).

Para programar una tarea debemos modificar el archivo crontab que se encuentra dentro del directorio /etc. Ya dentro del archivo se le pasa la frecuencia con la que se quiere programar la tarea y el comando a ejecutar, en nuestro caso el script que ejecutará.[4] [8]

```
SerSamMar jue dic 01 cat cuestion2.sh
#!/bin/bash

fecha=`date +%d-%m-%Y`
cp -r ~/codigo ~/seguridad/$fecha
SerSamMar jue dic 01 _
```

Figura 2.1: Script que se ejecuta en crontab diariamente

Con ese script lo que hacemos es copiar los archivos que hay en el directorio /codigo en el directorio /seguridad/fecha el cual diariamente irá cambiando ya que cada directorio creado tendrá la fecha del día que se ha hecho la copia.

Por último editamos el archivo crontab.

```
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 )
17 7 * * * root    /home/sersammar/cuestion2.sh
#
```

Figura 2.2: Muestra del contenido de crontab

Como se muestra en la figura, hemos añadido la última línea para que ejecute el script mostrado anteriormente todos los días a las 7:17

Y por último reiniciar el servicio para que los cambios surjan efecto.

3. Cuestión 3

3.1. Pruebe a ejecutar el comando, conectar un dispositivo USB y vuelva a ejecutar el comando. Copie y pegue la salida del comando. (considere usar `dmesg | tail`). Comente qué observa en la información mostrada.

Lo que he hecho ha sido ejecutar el comando `dmesg` sin ningún dispositivo conectado y lo he guardado en el archivo antes. Después he conectado el dispositivo y he vuelto a ejecutar el comando guardándolo en el archivo después.

Por último muestro por pantalla la diferencia entre los dos archivos, la cual vemos que es debido a la conexión de un dispositivo USB, en concreto mi smartphone de marca Huawei como se ve en la figura.

```
SerSamMar jue dic 01 dmesg > antes
SerSamMar jue dic 01 dmesg > despues
SerSamMar jue dic 01 diff antes despues
1164a1165,1170
> [ 1582.657032] usb 3-1: new high-speed USB device number 5 using xhci_hcd
> [ 1582.841953] usb 3-1: New USB device found, idVendor=12d1, idProduct=1051
> [ 1582.841962] usb 3-1: New USB device strings: Mfr=2, Product=3, SerialNumber
=4
> [ 1582.841967] usb 3-1: Product: ALE-L21
> [ 1582.841971] usb 3-1: Manufacturer: HUAWEI
> [ 1582.841975] usb 3-1: SerialNumber: QLF7N15702003881
SerSamMar jue dic 01
```

Figura 3.1: Muestra del contenido de crontab

4. Cuestión 4

- 4.1. Ejecute el monitor de "System Performance" y muestre el resultado. Incluya capturas de pantalla comentando la información que aparece.

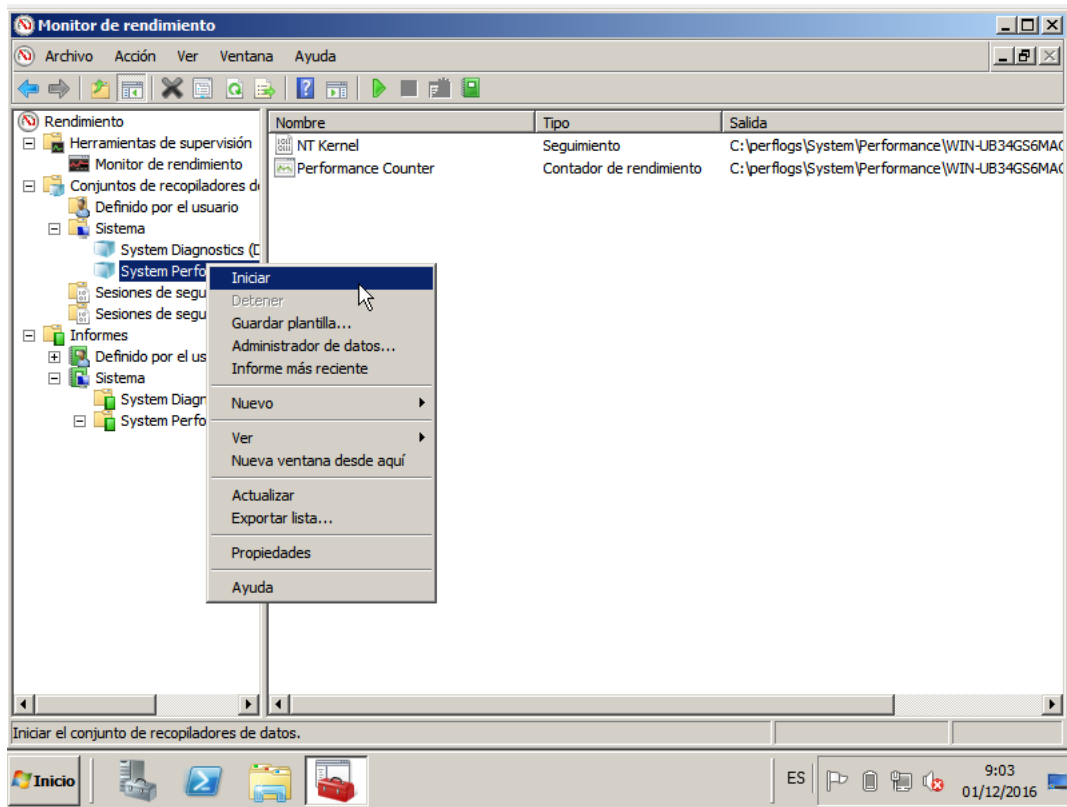


Figura 4.1: Inicio del System Performance

Primero hemos ejecutado el "System Performance" sin nada abierto, por lo que sabemos que el uso de CPU del sistema va a ser muy bajo.

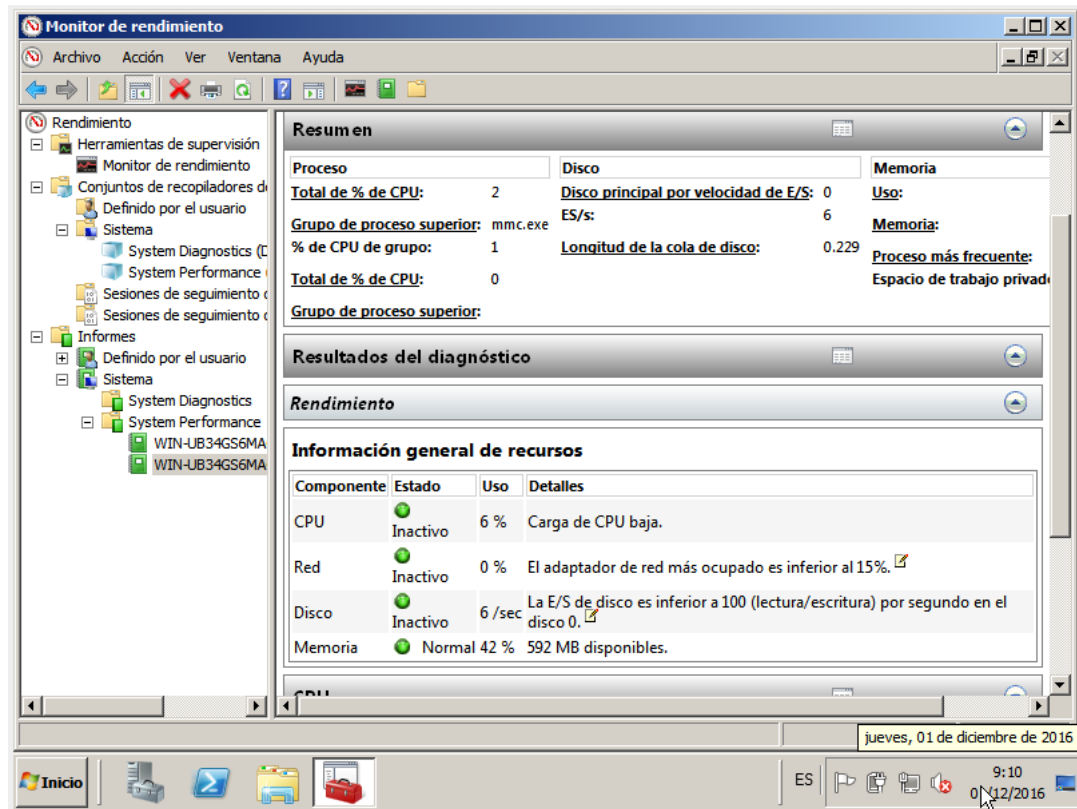


Figura 4.2: Contenido del informe del sistema con todos los programas cerrados

Ahora para hacer una pequeña comparación hemos abierto muchas ventanas de Internet Explorer, así como otras cuantas del bloc de notas.



Figura 4.3: Ventanas abiertas para hacer el System Performance

Una vez abierto todo esto, vamos a ejecutar de nuevo el "System Performance" a analizar los resultados.

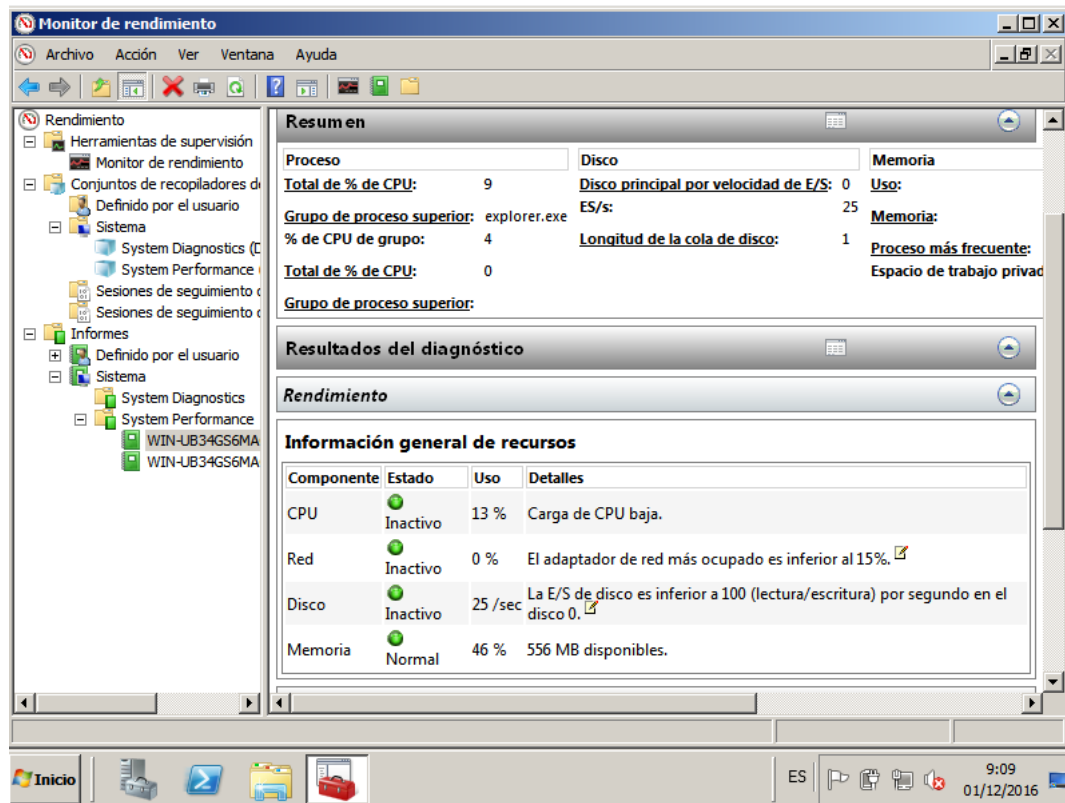


Figura 4.4: Muestra del contenido de crontab

Como podemos ver, el porcentaje de uso del CPU aumenta con respecto al primer informe sin nada abierto, esto es normal ya que tenemos más programas abiertos, así como el uso del disco y la memoria usada. Aún así, este informe general del sistema nos muestra que la carga de éste sistema está siendo baja, cosa que es totalmente lógica ya que no tenemos ningún programa pesado que nos esté consumiendo recursos.

5. Cuestión 5

5.1. Cree un recopilador de datos definido por el usuario (modo avanzado) que incluya tanto el contador de rendimiento como los datos de seguimiento:

Todos los referentes al procesador, al proceso y al servicio web.

Intervalo de muestra 15 segundos.

Almacene el resultado en el directorio Escritorio/logs.

Incluya las capturas de pantalla de cada paso.

Para crear un recopilador de datos que hayamos definido nosotros tendremos que crearlo, para ello pinchamos con el botón derecho Nuevo->Conjunto de recopilador de datos.

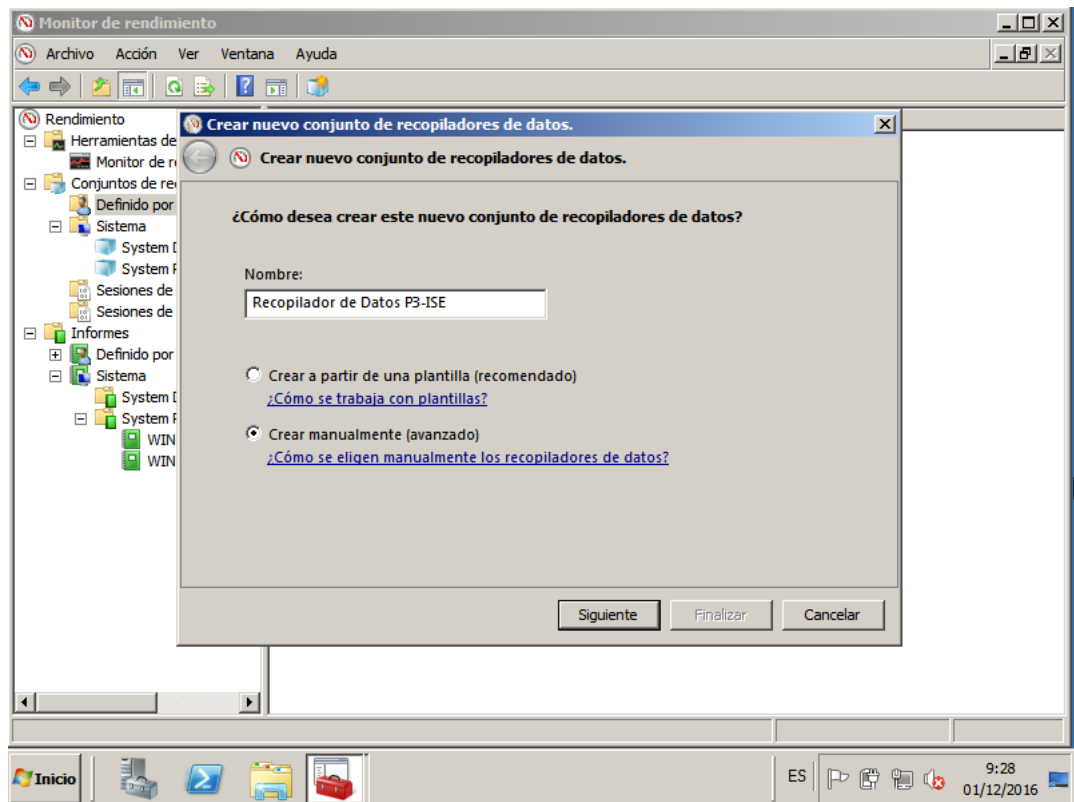


Figura 5.1: Define el nombre del recopilador de datos

Hecho esto daremos a Siguiente, y nos pedirá los datos que deseamos incluir.

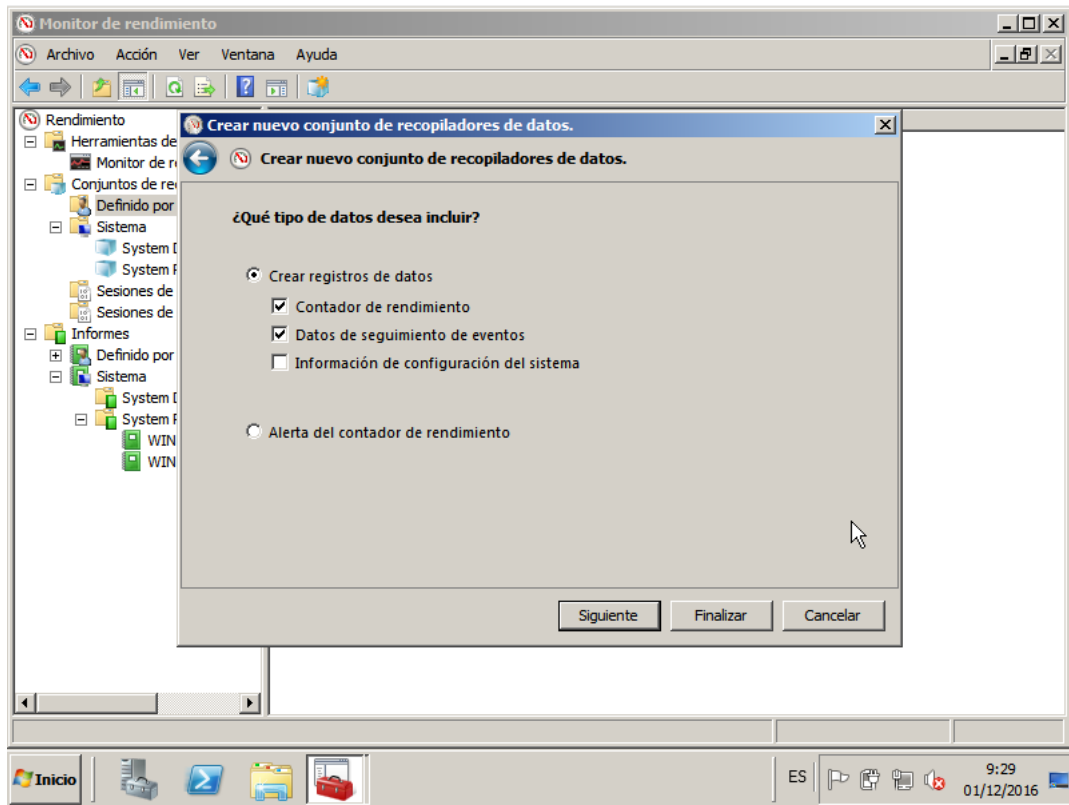


Figura 5.2: Incluir contador de rendimiento y datos de seguimiento

El paso siguiente será añadir los contadores que queremos, que en nuestro caso son los contadores del procesador, proceso y servicio web.

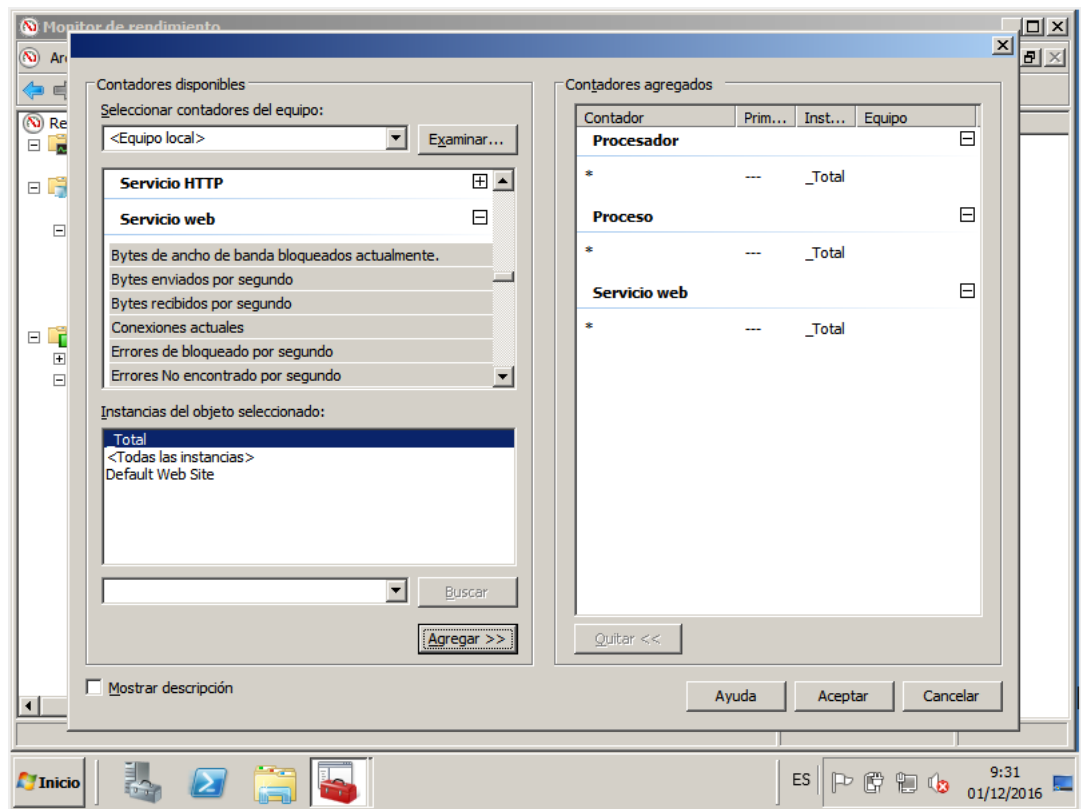


Figura 5.3: Añadiendo los contadores

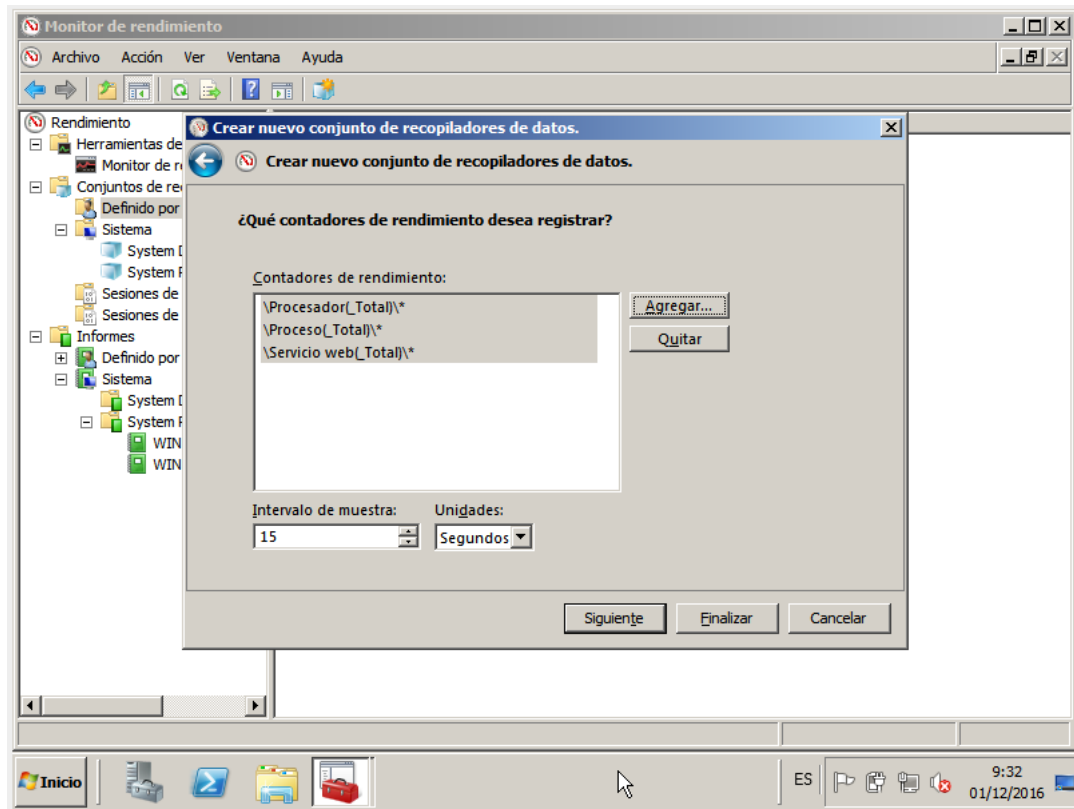


Figura 5.4: Definiendo el intervalo de muestreo

Damos a siguiente y añadimos los proveedores de seguimiento que deseamos.

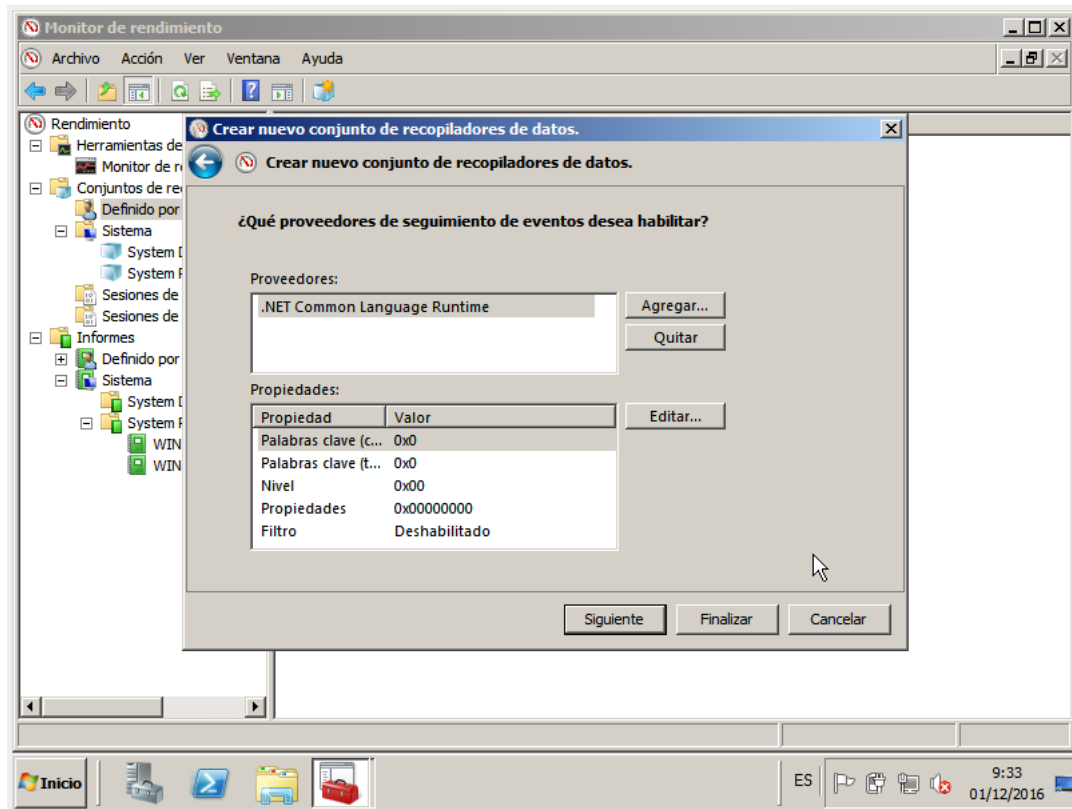


Figura 5.5: Proveedor añadido

Paso siguiente es indicarle dónde vamos a guardar los datos, en nuestro caso en Desktop/logs

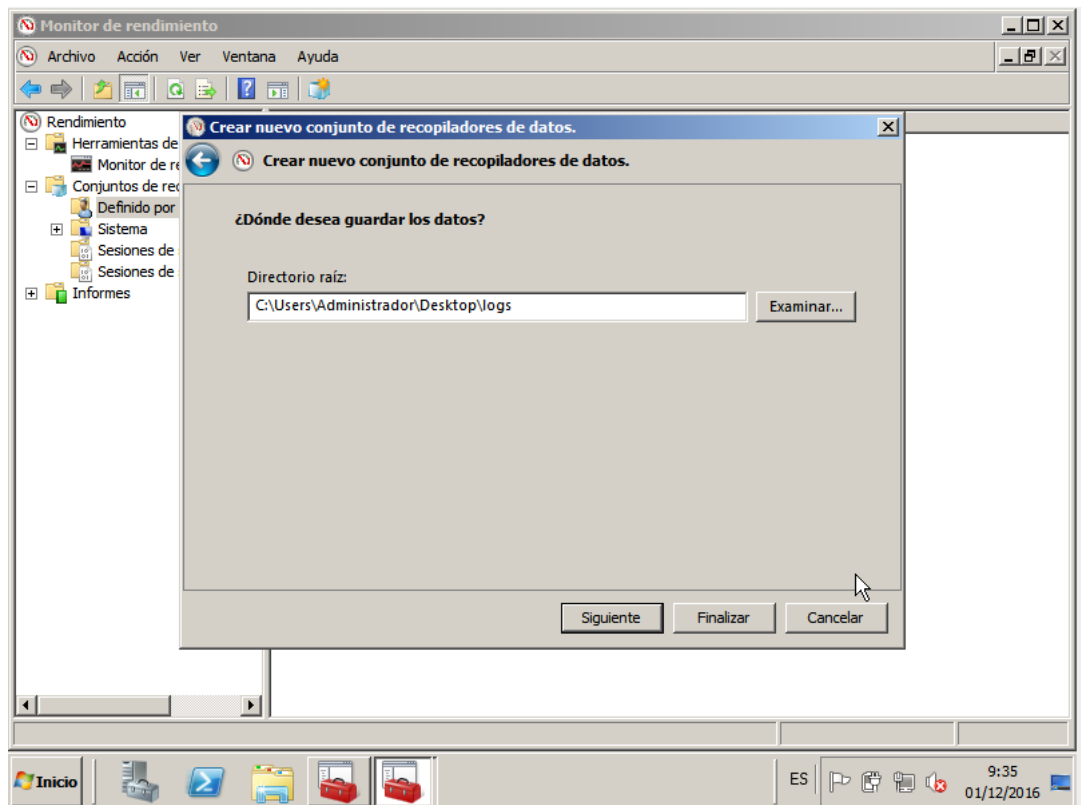


Figura 5.6: Ruta donde guardaremos los datos

Ahora ya tenemos creado nuestro propio recopilador de datos. Para mostrarlo lo iniciaremos y veremos los datos recopilados.

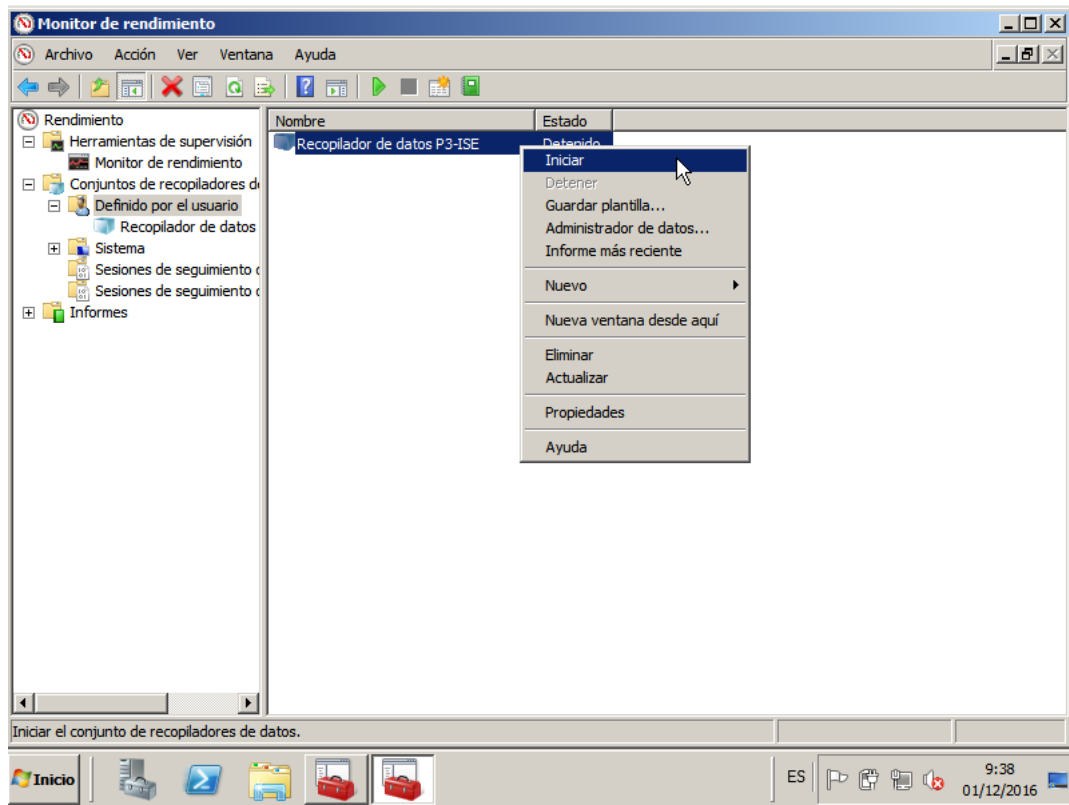


Figura 5.7: Iniciamos nuestro recopilador de datos

Una vez iniciado, nos vamos a la carpeta que indicamos anteriormente, en nuestro caso /Desktop/logs donde tendremos nuestro archivo de monitor.

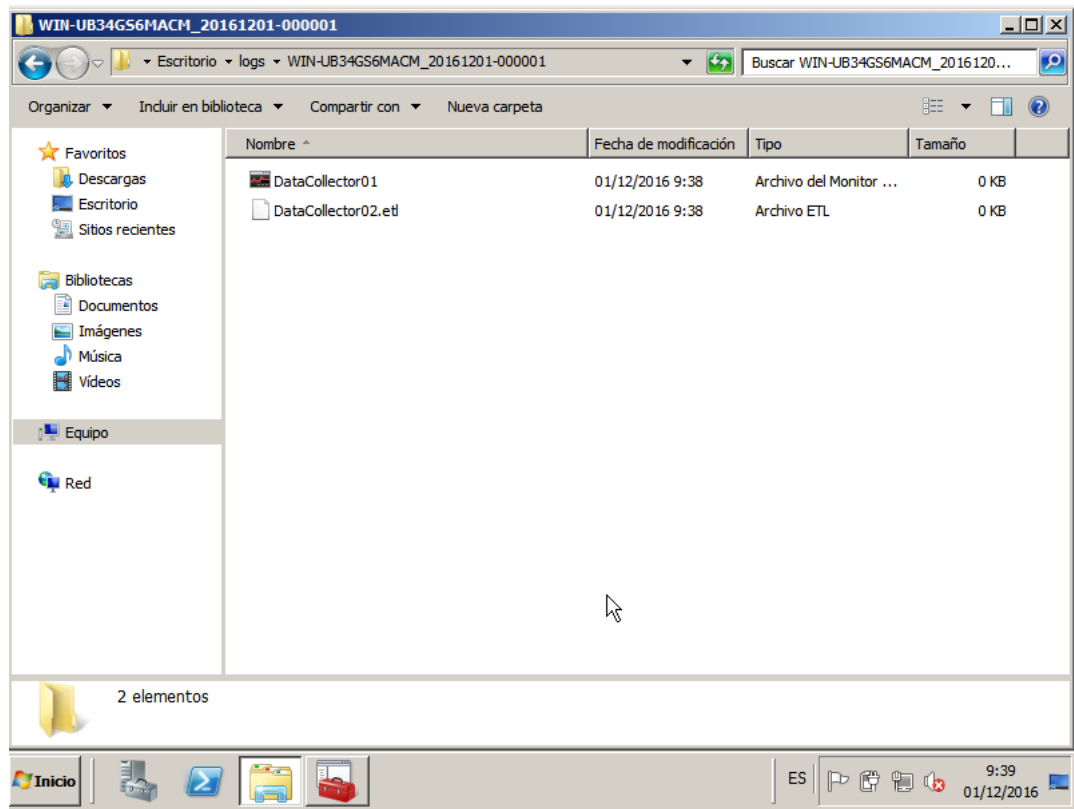


Figura 5.8: Archivos de /Desktop/logs

Por último, abrimos el archivo y nos mostrará la ejecución del monitor con los datos que se le indicó.

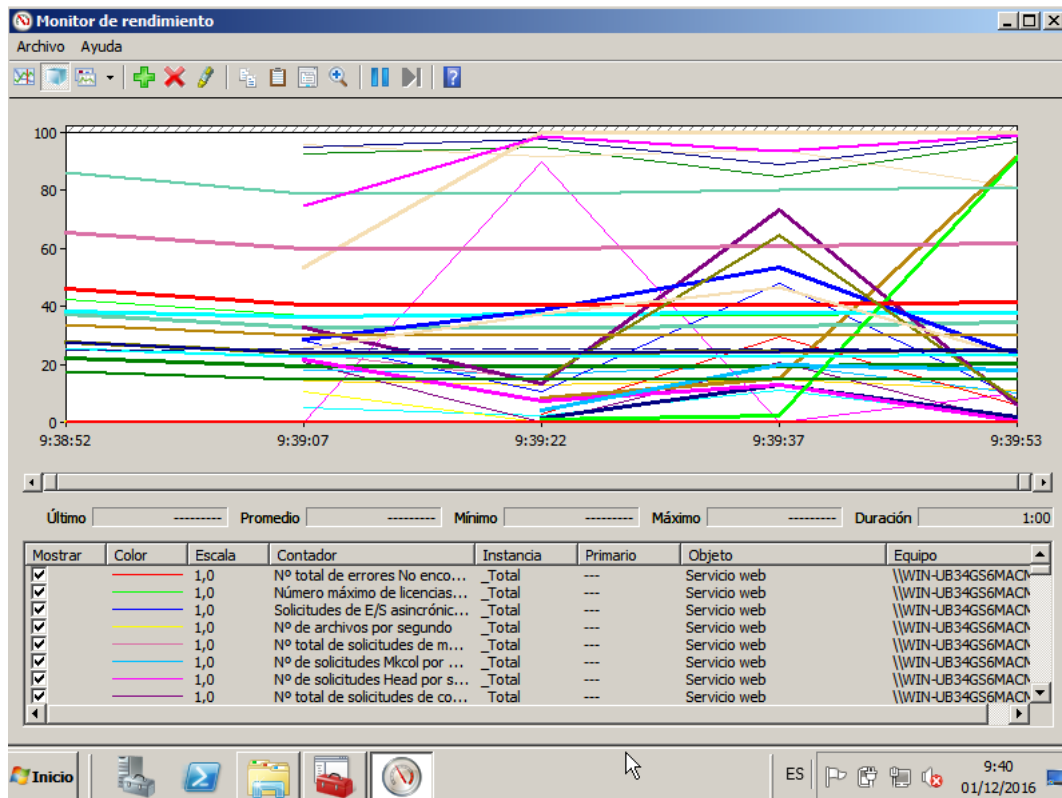


Figura 5.9: Archivo de monitor.

6. Cuestión 6

- 6.1. Visite la web del proyecto y acceda a la demo que proporcionan (<http://demo.munin-monitoring.org/>) donde se muestra cómo monitorizan un servidor. Monitoree varios parámetros y haga capturas de pantalla de lo que está mostrando comentando qué observa.

El primer parámetro que vamos a monitorizar es el uso de red por día.

Graph zoom

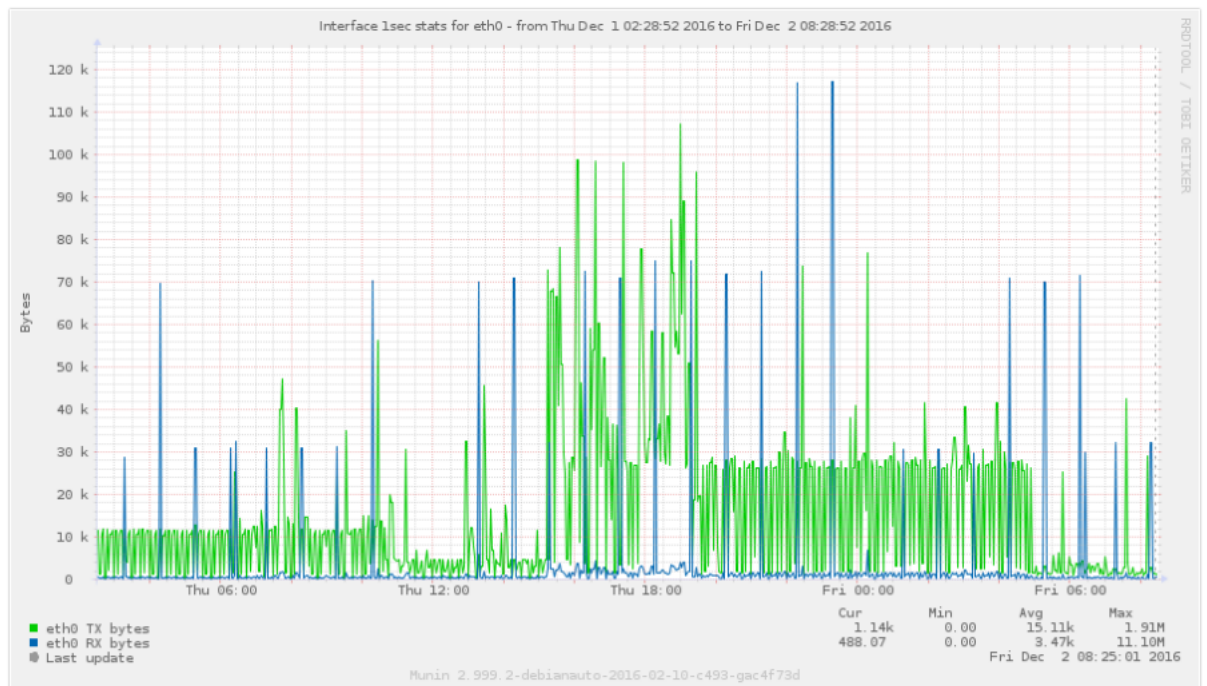


Figura 6.1: Monitorización de red.

Como podemos ver, el mayor uso de la red se encuentra en las horas de la tarde, sobre las 18:00 más o menos. Esto se debe a que son las horas del fin de la jornada de trabajo, y donde todos los empleados se conectan a la red para mandar correos, archivos, o el trabajo que hayan realizado. De ahí que estas horas el uso de red sea mucho más elevado que el resto.

Otro parámetro que vamos a ver será el uso de cpu.

Graph zoom

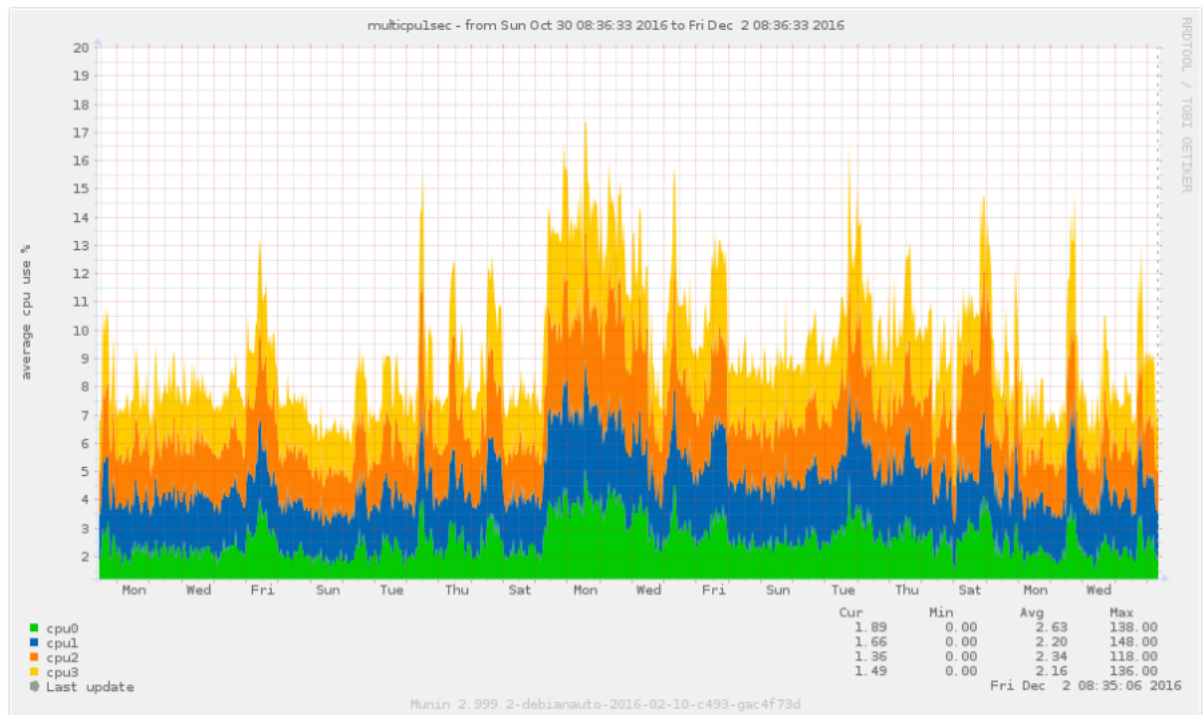


Figura 6.2: Monitorización de la cpu.

El uso de cpu lo vamos a monitorizar mensualmente, obteniendo el gráfico anterior. En cuanto a los posibles comentarios de este gráfico, cabe destacar que los inicios y finales de las semanas como son domingos y lunes, el uso de cpu se dispara, por eso surgen los picos del gráfico, esto es posible que se deba a que durante el fin de semana, los empleados no hacen uso del servidor, pero las primeras horas de la semana y las últimas, estos necesitan conectarse y lo hacen casi todos a la vez (debido a que tengan un horario y se conectan al empezar y antes de finalizar la jornada), el resto de días el trabajo puede estar más repartido y los empleados no tienen la necesidad de conectarse y desconectarse casi todos a la vez. Además se ve notablemente que hay una semana, en mitad del mes en la que el uso de la cpu es mayor que el resto de las semanas, ya que cuando esta acaba, disminuye bastante el uso de la cpu.

7. Cuestión 7

7.1. Escriba un breve resumen sobre alguno de los artículos donde se muestra el uso de strace o busque otro y coméntelo.

El artículo que vamos a comentar va a ser el que se encuentra en la url de la referencia.[1] Este artículo habla sobre la experiencia de un empleado con strace, al cual llamaron de una empresa para arreglar un problema de un servidor el cual no permitía acceder a algunos archivos de la base de datos.

Después de esta pequeña experiencia nos habla de strace y de su uso. Basándose en un ejemplo, nos explica el uso de esta herramienta, y de sus flags. Una vez explicado su uso, pasa a hacer uso de strace sobre un ejemplo real, donde se da cuenta que el proceso que está siguiendo hace uso de TCP y del socket de UNIX. Después de esto hace una comparación del proceso con el error y de el proceso sin el error, de tal forma que puede comparar entre ellos dónde está la mayor pérdida de tiempo.

Finalmente consigue ver que el error, que hace el proceso más lento estaba en un "log", y nos pone como ejemplo un código que usa la función sleep, y nos muestra cómo así podemos ver dónde está la función que gasta más tiempo, debido a que te muestra al final de cada sentencia el tiempo usado por esta.

Este artículo es bastante interesante, ya que nos explica el uso de strace, así como de algunas de las funcionalidades que presenta con sus flags, y además se presenta sobre un caso práctico y real, lo cual te motiva más a la hora de leerlo.

8. Cuestión 8

8.1. Escriba un script en Python o PHP y analice su comportamiento usando el profiler presentado.

El Script que vamos a realizar va a ser un Script sencillo en el que imprimiremos por pantalla dos frases, y entre ellas vamos a poner un sleep de 2 segundos.

El sleep se lo vamos a añadir para ver los tiempos de cada función ya que las dos funciones de print tardarán en realizarse 0 segundos.

```
#!/usr/bin/python
import time

print("Script para realizar el ejercicio 8 de ISE");
time.sleep(2);
print("El script de esta práctica lo vamos a realizar con Python");
```

Figura 8.1: Script para ejecutar el profiler sobre él.

Una vez hecho el script ejecutaremos sobre él el profiler, y en el caso de Python usaremos la función cProfile.

```
SerSamMar mar dic 06 python -m cProfile -s cumtime python.py
Script para realizar el ejercicio 8 de ISE
El script de esta práctica lo vamos a realizar con Python
3 function calls in 2.002 seconds

Ordered by: cumulative time

ncalls  tottime  percall  cumtime  percall filename:lineno(function)
      1   0.000    0.000    2.002    2.002 python.py:2(<module>)
      1   2.002    2.002    2.002    2.002 {time.sleep}
      1   0.000    0.000    0.000    0.000 {method 'disable' of '_lsprof.Prof
iler' objects}
```

Figura 8.2: Ejecución del profiler.

Como vemos, para ejecutar el script hacemos uso de cProfile, y además le he añadido cumTime para que vaya acumulando los tiempos de cada función.

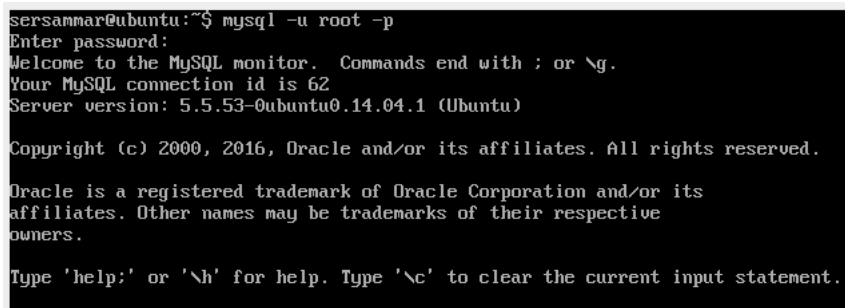
Como vemos las dos funciones de print tardan 0 segundos, y entre ellas está la función sleep que es la que hace que el script tarde los 2 segundos en terminar de ejecutarse.

Para conocer el uso de la función cProfile he hecho uso de la página referenciada,[7] en la que nos hace una introducción a los profilers de python.

9. Cuestión 9

9.1. Acceda a la consola mysql (o a través de phpMyAdmin) y muestre el resultado de mostrar el "profile" de una consulta (la creación de la BD y la consulta la puede hacer libremente).

Para acceder a mysql primero introducimos nuestra contraseña, lo cual indicamos con el flag -p, así nos pedirá la contraseña.



```
sersammar@ubuntu:~$ mysql -u root -p
Enter password:
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 62
Server version: 5.5.53-0ubuntu0.14.04.1 (Ubuntu)

Copyright (c) 2000, 2016, Oracle and/or its affiliates. All rights reserved.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.
```

Figura 9.1: Inicio de sesión de MySQL.

Una vez iniciada la sesión, lo que vamos a hacer es, lo primero activar el profiling, que por defecto está desactivado, para ello hacemos uso del comando `SET PROFILING = 1;` [2]

Una vez activado el profiling, haremos una serie de consultas sobre la base de datos, la primera será crear una base de datos, y después añadirle una tabla. Lo cual se muestra cómo hacerlo en la página que se cita a continuación, donde hay una pequeña introducción a SQL [5]

Por último, mostraremos los tiempos de cada sentencia haciendo uso del comando `SHOW PROFILES;` [3]

```

mysql> SET PROFILING = 1;
Query OK, 0 rows affected (0.00 sec)

mysql> SHOW PROFILES;
Empty set (0.00 sec)

mysql> CREATE DATABASE pruebaISE;
Query OK, 1 row affected (0.00 sec)

mysql> USE pruebaISE;
Database changed
mysql> CREATE TABLE ise(nombre VARCHAR(40));
Query OK, 0 rows affected (0.01 sec)

mysql> SHOW PROFILES;
+-----+-----+-----+
| Query_ID | Duration | Query |
+-----+-----+-----+
| 1 | 0.00074150 | CREATE DATABASE pruebaISE |
| 2 | 0.00019150 | SELECT DATABASE() |
| 3 | 0.00052875 | show databases |
| 4 | 0.00027875 | show tables |
| 5 | 0.01334975 | CREATE TABLE ise(nombre VARCHAR(40)) |
+-----+-----+-----+
5 rows in set (0.00 sec)

mysql>

```

Figura 9.2: Inicio de sesión de MySQL.

Referencias

- [1] <http://chadfowler.com/2014/01/26/the-magic-of-strace.html>.
- [2] <http://dev.mysql.com/doc/refman/5.7/en/show-profile.html>.
- [3] <http://dev.mysql.com/doc/refman/5.7/en/show-profiles.html>.
- [4] <http://man7.org/linux/man-pages/man8/cron.8.html>.
- [5] <http://mysql.conclase.net/curso/?cap=007>.
- [6] <https://debian-handbook.info/browse/es-ES/stable/sect.manipulating-packages-with-dpkg.html>.
- [7] <https://docs.python.org/2/library/profile.html>.
- [8] <https://linux.die.net/man/5/crontab>.
- [9] <http://www.estrellateyarde.org/logs-en-linux>.