

# LABORATORIO 3

## 1. GESTIÓN DE RECURSOS DEL SISTEMA

1) Obtener el número de procesos en ejecución en el sistema.

```
nagore@instance-1:~$ ps -e | wc -l
```

2) Obtener el número de procesos en ejecución que pertenezcan a usuario root.

```
nagore@instance-1:~$ ps -U root | wc -l
```

3) Instalar el paquete “stress-ng”. Ejecutar el benchmark para 1 núcleo de CPU y 20 segundos.

```
nagore@instance-1:~$ stress-ng --cpu 1 --timeout 20s
```

4) Ejecutar de nuevo “stress-ng”, esta vez sin límite de tiempo. Mientras esté en ejecución:

- a. Usar una señal para pausar su ejecución.
- b. Usar una señal para reanudar su ejecución.
- c. Reducir la prioridad del proceso al mínimo. ¿Cambia algo?

```
nagore@instance-1:~$ stress-ng --cpu 1
```

```
nagore@instance-1:~$ pgrep stress-ng
```

```
1078
```

```
1079
```

```
nagore@instance-1:~$ kill -SIGSTOP 1078
```

```
nagore@instance-1:~$ kill -SIGCONT 1078
```

```
nagore@instance-1:~$ pgrep stress-ng
```

```
1078
```

```
1079
```

```
nagore@instance-1:~$ renice 19 -p 1078
```

**19 prioridad minima, -20 prioridad maxima**

5) Detectar qué proceso tiene la mayor prioridad en el sistema. Buscar cuál es el propósito de ese proceso.

```
nagore@instance-1:~$ ps -eo pid,ni,cmd --sort=ni | head -n 2
```

Este proceso se llama "khugepaged" y es parte del kernel del sistema Linux. Su propósito principal es administrar y gestionar el proceso de gestión de páginas grandes en la memoria del sistema.

Para ordenar por prioridad, vamos a entrar en top y vamos a pulsar la tecla f. Con esta, se nos abrirán todos los campos que muestra y debemos ir a prioridad y pulsar s. Una vez hecho esto, salimos usando q y top quedará ordenado por prioridad. Ahora, pulsamos la tecla R para ordenar de mayor a menor prioridad.

Los procesos que más prioridad tienen en el sistema son migration/0, migration/1 y multipathd. Los procesos migration gestionan los módulos de la CPU. Por lo tanto, nuestra CPU tiene dos módulos. El proceso multipathd sirve para conectarse con la memoria del sistema.

**6) Limitar el máximo tiempo de uso de CPU a 5 minutos para todos los usuarios.**

```
# Comprobamos los límites actuales  
nagore@instance-1:~$ ulimit -a
```

```
# Límitamos el tiempo de todos los usuarios a 5 minutos (5min = 300s)  
nagore@instance-1:~$ ulimit -t 300
```

```
# Comprobamos el límite establecido  
nagore@instance-1:~$ ulimit -a
```

**7) Crear un fichero crontab para el usuario root con las siguientes tareas:**

- a. Ejecutar el comando date cada minuto y escribir su salida estándar al fichero /tmp/date.log (se debe escribir al final del fichero cada vez).
- b. Borrar el directorio /tmp los primeros 5 días de cada mes a las 17:00.

```
nagore@instance-1:~$ sudo crontab -u root -e  
* * * * * date >> /tmp/date.log  
0 17 1-5 * * rm -rf /tmp
```

>> si desea agregar datos al archivo de registro existente sin borrar su contenido actual o, >, si desea sobrescribir completamente el archivo con el último registro

**8) Comprobar que las tareas de cron funcionan correctamente (ver el fichero /tmp/date.log).**

```
nagore@instance-1:~$ cat /tmp/date.log
```

**Las siguientes tareas del laboratorio son relativas a las conexiones de red y se propone trabajar en parejas. Seréis A y B, y cada uno utilizará su propia máquina virtual:**

**9) Usando netcat, A abre una conexión a la escucha en el puerto 3000. Si fuese necesario abrir puertos de Google Cloud, se recomienda este tutorial 1. B se conecta a ese puerto y**

**escribe el mensaje “Hola A”. Tras recibirla, en la misma conexión, B escribe “Hola B” y cierra la conexión.**

- A (Sergio):

```
nc -l -p 3000
```

- B (Yo):

```
nc 35.224.232.153 3000
```

**10) El comando “dd if=/dev/urandom” sirve para generar números aleatorios. A abre una conexión a la escucha en el puerto 3000 y B envía números aleatorios a esa conexión de A. Sin cerrar la conexión, A y B utilizan nethogs para comprobar la tasa de bytes enviados y recibidos en cada parte. ¿Los valores coinciden? Después, A cierra la conexión.**

- A (Sergio):

```
nc -l -p 3000
```

- B (Yo):

```
dd if=/dev/urandom | nc 35.224.232.153 3000
```

- Ambos, desde otra terminal:

```
sudo nethogs
```

**11) A crea localmente un fichero con texto aleatorio. Utilizar netcat para que A envíe este fichero a B. ¿Qué diferencia hay entre utilizar netcat y scp para enviar ficheros entre diferentes máquinas?**

- A (Sergio):

```
nc -l -p 3000 > archivo.txt
```

- B (Yo):

```
nc 35.224.232.153 3000 < archivo.txt
```

En resumen, la principal diferencia es que netcat se utiliza para transmitir datos en tiempo real y requiere comandos personalizados para la transferencia, mientras que scp se utiliza para transferir archivos completos de manera segura y simple. Si la seguridad y la integridad del archivo son importantes, scp es la opción más recomendada. Por otro lado, si solo necesitas transmitir datos en tiempo real sin preocuparte por la transferencia completa de archivos, netcat podría ser útil.

## 2. GESTIÓN DE LOS REGISTROS DEL SISTEMA

**1) Leer la página de manual de logger. ¿Con qué parámetro se indica la prioridad de los mensajes?**

-p o –priority, para especificar la prioridad del mensaje

**2) Utilizando la línea de comandos, enviar el mensaje “Hola Mundo de Logs” al fichero /var/log/syslog. Comprobar que se ha hecho correctamente.**

```
nagore@instance-1:~$ sudo logger "Hola Mundo de Logs"
```

```
nagore@instance-1:~$ tail /var/log/syslog
```

**3) Enviar todos los mensajes de nivel “debug” generados por el servicio sshd al fichero /var/log/ssh.log. Este fichero debe haber sido creado anteriormente y estar vacío. Configurar el servicio sshd para que funcione en modo “debug” y comprobar el efecto que tiene en el fichero ssh.log.**

```
nagore@instance-1:~$ sudo touch /var/log/ssh.log
```

```
nagore@instance-1:~$ sudo nano /etc/ssh/sshd_config
```

```
LogLevel DEBUG
```

```
nagore@instance-1:~$ sudo nano /etc/rsyslog.d/50-default.conf
```

```
auth.debug -/var/log/ssh.log
```

```
nagore@instance-1:~$ sudo service rsyslog restart
```

```
nagore@instance-1:~$ sudo systemctl restart sshd
```

```
nagore@instance-1:~$ sudo chown syslog:adm /var/log/ssh.log
```

```
nagore@instance-1:~$ cat /var/log/ssh.log
```

**4) Configurar la rotación de logs (fichero /var/log/syslog) para que se guarden de manera mensual y comprimida, y para que todos los logs generados en un año se guarden en un directorio llamado /var/log/syslog.old.**

```
nagore@instance-1:~$ sudo nano /etc/logrotate.d/rsyslog
```

```
/var/log/syslog {
```

```
    monthly
```

```
    compress
```

```
    create 0644 root root # Permisos para el nuevo archivo de log
```

```

rotate 12      # Mantener hasta 12 archivos de log antiguos
dateext       # Agregar fecha a los archivos rotados
dateformat -%Y-%m-%d # Formato de fecha para los archivos rotados
olddir /var/log/syslog.old/ # Directorio para archivos antiguos
}

/var/log/syslog{
    rotate 12
    monthly
    compress
    olddir /var/log/syslog.old
    createolddir 0640 syslog adm
}

```

nagore@instance-1:~\$ sudo logrotate -d /etc/logrotate.d/syslog

### 3. MONITORIZACIÓN EN GOOGLE CLOUD PLATFORM

**0) Enviad un mensaje al log del sistema desde la Shell y verificar que se visualiza en el panel de este nuevo Dashboard**

nagore@instance-1:~\$ gcloud logging write PRUEBA "Mensaje de prueba"

**1) Comprobar cuánto espacio libre hay en la partición /dev/sda1 de vuestra máquina virtual.**

nagore@instance-1:~\$ df -h -->30%

**2) Configurad una alerta en Google Cloud para que se os envíe un e-mail cuando el espacio en la partición /dev/sda1 de vuestra máquina virtual sea un 5% mayor del actual (p.e. si ahora mismo está usado un 15%, cuando supere el 20%).**

**3) Utilizando el comando dd cread uno (o varios) ficheros en la carpeta /tmp de vuestra máquina virtual con datos aleatorios. El tamaño de estos ficheros debe hacer que el uso de partición se incremente hasta superar el umbral establecido en el paso anterior.**

#archivo de 3GB (50MB --> bs=1M count 50)

nagore@instance-1:~\$ dd if=/dev/zero of=/tmp/archivo\_aleatorio bs=1G count=3

nagore@instance-1:~\$ df -h --> 40%

**4) Verificad que el incremento de consumo en disco se refleja en el Dashboard creado en**

el paso anterior.

No verificado

**5) Verificad que habéis recibido uno (o varios) correos de Google Cloud con la alerta.**

Verificado

**6) Eliminad los ficheros creados en el paso 3 y la alerta creada en el paso 2.**

Ficheros eliminados

Alerta desactivada

## 4. EVALUACIÓN DE RENDIMIENTO

**1) Descargar el benchmark desde la web oficial: <https://tu-dresden.de/zih/firestarter>**

```
(base) nagore@nagore-ThinkPad:~/Deskargak$ tar -xzvf FIRESTARTER_2-0-tar.gz
```

```
(base) nagore@nagore-ThinkPad:~/Deskargak$ ./FIRESTARTER
```

**2) Ejecutarlo durante 30 segundos, haciendo que muestre información adicional (un “report” de rendimiento). Buscar los parámetros necesarios para ello. ¿Qué valor de GFLOP/s obtienes? Este valor es una métrica de la capacidad de cómputo de tu CPU.**

```
(base) nagore@nagore-ThinkPad:~/Deskargak$ ./FIRESTARTER -t 30 -r
```

**3) La mayoría de CPUs modernas tienen varios conjuntos de instrucciones que permiten obtener diferentes niveles de rendimiento. Firestarter es capaz de detectar los conjuntos de instrucciones disponibles en la CPU y usarlos para evaluar su capacidad. Encuentra el parámetro que muestra los tipos disponibles en tu CPU.**

```
(base) nagore@nagore-ThinkPad:~/Deskargak$ ./FIRESTARTER --list-instruction-groups
```

**4) Crea un script que ejecute Firestarter con cada tipo de instrucción disponible en tu CPU durante 30 segundos y reporte el mayor valor de GFLOP/s obtenido. Esté será el mayor rendimiento que el benchmark es capaz de obtener en tu CPU.**

```
#!/bin/bash
```

```
tipos_de_instrucciones=("L1_2L" "L1_2LS_256" "L1_L" "L1_LS" "L1_LS_256" "L1_S"  
"L2_2LS_256" "L2_L" "L2_LS" "L2_LS_256" "L2_S" "L3_L" "L3_LS" "L3_LS_256" "L3_P" "L3_S"  
"RAM_L" "RAM_LS" "RAM_P" "RAM_S")
```

```

mayor_gflop=0

# Itera a través de los tipos de instrucciones

for tipo_instruccion in "${tipos_de_instrucciones[@]}"; do

    echo "-----"
    echo "Ejecutando Firestarter con instrucciones $tipo_instruccion durante 30
segundos..."

    # Comando para ejecutar Firestarter con el tipo de instrucción específico
    ./FIRESTARTER -t 30 -r --run-instruction-groups $tipo_instruccion:1 >out4.txt

    # Extraer el valor de GFLOP/s del informe de rendimiento (ajusta según la salida de
    Firestarter)

    gflop_actual=$(grep -oP 'estimated floating point performance: \K[0-9.]+(?=GFLOPS)' out4.txt)

    echo "-----"
    echo "GFLOPS de la instruccion $tipo_instruccion es $gflop_actual"
    echo "-----"

    # Comprobar si el valor actual es mayor que el mayor valor registrado

    if (( $(echo "$gflop_actual > $mayor_gflop" | bc -l) )); then
        mayor_gflop=$gflop_actual
    fi

done

# Reportar el mayor valor de GFLOP/s obtenido

echo "El mayor valor de GFLOP/s obtenido es: $mayor_gflop"

```

**5) Crea un script similar al anterior, pero que devuelva el máximo valor obtenido para el ancho de banda de memoria (cuántos GB/s).**

```

#!/bin/bash

tipos_de_instrucciones=("L1_2L" "L1_2LS_256" "L1_L" "L1_LS" "L1_LS_256" "L1_S"
"L2_2LS_256" "L2_L" "L2_LS" "L2_LS_256" "L2_S" "L3_L" "L3_LS" "L3_LS_256" "L3_P" "L3_S"
"RAM_L" "RAM_LS" "RAM_P" "RAM_S")

mayor_ab=0

# Itera a través de los tipos de instrucciones

```

```
for tipo_instrucion in "${tipos_de_instrucciones[@]}"; do
    echo "-----"
    echo "Ejecutando Firestarter con instrucciones $tipo_instrucion durante 30 segundos..."
    # Comando para ejecutar Firestarter con el tipo de instrucción específico
    ./FIRESTARTER -t 30 -r --run-instruction-groups $tipo_instrucion:1 >out5.txt
    # Extraer el valor de ancho de banda del informe de rendimiento (ajusta según la salida de Firestarter)
    ab_actual=$(grep -oP 'estimated memory bandwidth\*: \K[0-9.]+(?= GB/s)' out5.txt)
    echo "-----"
    echo "Ancho de banda de la instrucción $tipo_instrucion es $ab_actual"
    echo "-----"
    # Comprobar si el valor actual es mayor que el mayor valor registrado
    if (( $(echo "$ab_actual > $mayor_ab" | bc -l) )); then
        mayor_ab=$ab_actual
    fi
done
echo "El mayor valor de ancho de banda obtenido es: $mayor_ab"
```