

EXAMEN PRÁCTICO. CONVOCATORIA SEPTIEMBRE 2020/2021

Nombre y Apellidos: _____ Alu: _____

1. Realizar un shell script en BASH con las siguientes funcionalidades:

- a) Mostrar un listado con todos los usuarios que tienen al menos un proceso en ejecución, indicando el porcentaje de memoria utilizada por todos los procesos de ese usuario.

Para ello sumaremos el porcentaje de memoria utilizado por cada proceso propiedad del usuario. (comando bc)

El listado está ordenado alfabéticamente por nombre de usuario, e incluirá el nombre del usuario, UID del usuario, el GID, el número de procesos totales del usuario y el porcentaje total de memoria utilizada.

3 PUNTOS

- b) Con la opción -inv, se ordenará inversamente el nombre de usuario. **1 PUNTO**
- c) Con la opción -nodaemon, se mostrará solo los usuarios reales que están conectados en el sistema. **1 PUNTO**

Todos los parámetros se pueden utilizar de forma simultánea.

2. Realizar un programa en C o C++ que envíe por un socket UDP el contenido de un archivo de cualquier tamaño y otro que lo reciba, escribiendo en el disco duro una copia exacta del original. La sintaxis debe ser:

```
send  [opciones]  [ARGUMENTO]
recv  [opciones]  [ARGUMENTO]
```

- a) Envío y recepción con nombres de archivos indicados como argumento de la línea de comandos. **3.5 PUNTOS**

```
send origen.txt
recv destino.txt
```

- b) La dirección IP y puerto de destino se indican con las opciones -d y -p de línea de comandos en el programa send y el puerto de escucha se indica con la opción -p de línea de comandos en el programa recv. **1.5 PUNTOS**

```
send -i 123.1.2.4 -p 8080 origen.txt
recv -p 8080 destino.txt
```

Como los programas debe funcionar con archivos de tamaño arbitrario, es interesante generar automáticamente estos archivos para probar los programas. Para eso se puede usar el siguiente comando:

```
dd if=/dev/urandom of=sample.txt bs=2M count=1
```

que genera un archivo de 2MB. Cambiando el valor asignado a la opción bs se pueden crear archivos de otros tamaños. Además, es importante comprobar que el archivo original y la copia son exactamente iguales, para lo que se puede usar el comando:

```
cmp origen.txt destino.txt
```

Para obtener el 100% de la nota se deben detectar y manejar errores de las llamadas al sistema para terminar el programa liberando previamente los recursos y evitar dejar hijos huérfanos y procesos zombie. Se deben utilizar solo llamadas al sistema (open, read, write) y sockets UDP para el envío.

NOTA: En ambas preguntas se valorará la claridad y sencillez a la hora de escribir el código y los comentarios en el mismo. Cualquier suposición sobre la práctica o modos de uso concretos debe ser especificada en el examen.