Práctica 1. Monitorización de procesos

En esta práctica veremos algunas herramientas para la monitorización de procesos. Utilizaremos una **máquina virtual**.

Contenido:

Creación de una máquina virtual para pruebas (~10 min.)

Arranca Linux y entra como "Usuario VMs". Introduce tu usuario y contraseña.

Abre la carpeta `Disco VMs` desde el escritorio y ve al directorio `ECO`. Pulsa dos veces en el fichero `ECO.ova` para abrirlo. Pulsa en "Importar" en la ventana de VirtualBox que aparecerá.

Desde VirtualBox, selecciona la máquina virtual "ECO" y pulsa en "Iniciar" para arrancarla. Entra con el usuario "usuario" y la contraseña "usuario".

time (~20 min.)

Instala el programa `time`:

```
$ sudo apt-get update
$ sudo apt-get install time
```

Consulta la página de manual de `time` y la información proporcionada por `help time` (`help` proporciona información sobre palabras reservadas y comandos internos de `bash`). También puedes buscar la palabra reservada `time` en la página de manual de `bash`.

Mide alguna orden con las dos alternativas (programa y palabra reservada) y observa las diferencias. Con la opción `-p` de ambas, se usa el formato de salida del estándar POSIX.

El programa `time` mide el tiempo de respuesta mediante `gettimeofday(2)`, ejecutada antes y después de ejecutar (con `fork(2)` y `execve(2)`) la orden a medir. El tiempo de procesador en modo usuario y sistema se obtiene con `wait3(2)` (implementada con `wait4(2)`), que espera a que el proceso hijo termine y devuelve una estructura `rusage`. Esta estructura se describe en la página de manual de `getrusage(2)` y está definida en `/usr/include/linux/resource.h`.

Prueba la opción `-v` del programa `time`.

Indica a qué campo de la estructura `rusage` se corresponde cada valor proporcionado por `time -v`. Indica, si lo sabes, cómo se obtienen los demás valores.

Mide los tiempos de ejecución de las siguientes órdenes (una a una):

```
$ find /usr > /dev/null   # primera vez (caches del FS vacías)
$ find /usr > /dev/null   # siguientes veces (caches llenas)
$ dd if=/dev/zero of=/var/tmp/prueba count=100K bs=1K
$ dd if=/dev/zero of=/var/tmp/prueba count=100K bs=1K
oflag=direct
```

```
$ dd if=/dev/urandom of=/var/tmp/prueba count=100K bs=1K
```

Para asegurarte de que estás ejecutando la primera orden `find`, con las *caches* del sistema de ficheros vacías, puedes vaciarlas con el siguiente comando:

```
$ sudo sysctl -w vm.drop_caches=3
```

Copia los resultados y escribe un breve análisis de los mismos, indicando si las tareas anteriores son limitadas por procesador (CPU-bound) o por E/S (IO-bound), en función de si pasan más tiempo usando el procesador o esperando por E/S.

ps (~10 min.)

Consulta la página de manual de `ps`.

Escribe un único comando que muestre el usuario, la prioridad, el porcentaje de uso del procesador y el tamaño de memoria virtual y física de todos los procesos del usuario `root`, ordenados de mayor a menor uso de memoria física.

Escribe el comando solicitado.

top (~30 min.)

Consulta la página de manual de `top`.

Ejecuta `top` y pulsa la tecla `h`. Prueba los distintos comandos interactivos que se indican.

Compila el programa `cpu_mem.c` (disponible con la práctica), añadiendo la opción `-lm` para enlazar con la biblioteca matemática.

Observa cómo evoluciona el tamaño de memoria virtual, el tamaño de memoria residente y el porcentaje de uso de procesador y memoria del proceso `cpu_mem` al ejecutar el siguiente comando:

```
$ ./cpu_mem 1200
```

donde el argumento es un valor ligeramente superior a la cantidad de memoria física total en MB (1024 en la máquina virtual). Si aparece el mensaje “Terminado (killed)”, significa que el *OOM (Out Of Memory) Killer* ha entrado en funcionamiento, por lo que deberás reducir su valor.

Observa también cómo evoluciona el porcentaje de procesador usado por `kswapd0` (*Kernel Swap Daemon*), que es el *thread* del *kernel* encargado de liberar páginas de memoria, cuando se ejecuta el comando anterior.

Para poder ver la evolución, es recomendable usar `top` con las opciones `-b` (*batch*) y `-d 1` (*delay*) y filtrar la información de los procesos mencionados con `egrep "PID|cpu_mem|kswapd0"`.

Copia los resultados y escribe un breve análisis de los mismos.

procfs (~20 min.)

Consulta la página de manual de `proc`.

Observa el contenido del directorio `/proc` y examina el contenido de los siguientes ficheros:

- `cpuinfo`
- `loadavg`
- `meminfo`
- `vmstat`
- `diskstats`
- `interrupts`
- `$$/status`
- `$$/sched`
- `$$/io`
- `$$/net/netstat`

Describe el contenido de los ficheros anteriores.