Práctica 1. Monitorización de procesos

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`. Abre el fichero `ECO.ova` haciendo doble *click*. Pulsa en “Importar” en la ventana de VirtualBox que aparecerá.

Desde VirtualBox, selecciona la máquina virtual “ECO” y pulsa en “Iniciar” para arrancarla. Entra como usuario “usuario”, con contraseña “usuario”.

procfs (~20 min)

Consulta la página de manual de `proc`.

Observa el contenido del directorio `/proc` (`cd /proc`) y examina el contenido de los siguientes ficheros (editores de texto: `nano`, `gedit`, `vi`, ...):

- `cpuinfo`
- `meminfo`
- `swaps`
- `loadavg`
- `diskstats`
- `vmstat`
- `interrupts`
- `$$/status`
- `$$/maps`
- `$$/limits`
- `$$/sched`
- `$$/io`
- `$$/net/dev`
- `$$/net/netstat`

time (~15 min)

Instala el programa `time`:

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

Consulta la página de manual de `time` y busca la palabra reservada `time` en la página de manual de `bash` (programa para interpretar órdenes compatible con POSIX).

Mide alguna orden con las dos alternativas y observa las diferencias. Con la opción `-p` de ambas, se usa el formato de salida del estándar POSIX (intérprete estándar de órdenes en linux).
El programa `time` mide el tiempo de respuesta mediante `gettimeofday(2)`, ejecutada antes y después de ejecutar (con `fork(2)` y `exec(3)`) la orden a medir. El tiempo de

CPU en modo usuario y sistema se obtiene con la `wait3(2)` (que llama a `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`.

Entrega: Crea una tabla asociando los valores proporcionados por `time -v` con los campos de la estructura `rusage`. Indica cómo se obtiene el resto de valores.

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

```
$ find /usr > /dev/null     # caches del FS vacías
$ find /usr > /dev/null
$ 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
```

Si quieres repetir la ejecución de `find` con las *caches* del sistema de ficheros vacías, puedes usar la siguiente orden para vaciarlas sin tener que reiniciar el sistema:

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

Entrega: Escribe un breve análisis de los resultados indicando si las tareas anteriores son limitadas por CPU (*CPU-bound*) o por E/S (*IO-bound*), en función de si pasa más tiempo usando la CPU o esperando por E/S.

ps (~5 min)

Consulta la página de manual de `ps`.

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

Entrega: Escribe el comando solicitado.

top (~25 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` (código fuente disponible en el Campus Virtual), añadiendo la opción `-lm` para enlazar usando las librerías matemáticas.

Observa cómo evoluciona el tamaño de memoria virtual, el tamaño de memoria residente y el porcentaje de CPU y memoria usados por el 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 MV). Si aparece el mensaje “Terminado (killed)”, significa que el *OOM (Out Of Memory) Killer* ha entrado en funcionamiento, por lo que deberás reducir este valor.

Observa cómo evoluciona el porcentaje de CPU usado por `kswapd0` (*Kernel Swap Daemon*), que es el *thread* del *kernel* encargado de liberar páginas de memoria.

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

Entrega: Escribe un breve análisis de los resultados. Las tablas de datos generadas en este apartado serán guardadas en un fichero `.csv`.