Práctica 2.1. 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 y examina el contenido de los siguientes ficheros:

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

# time (~20 min)

Instala el programa time:

$ 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.

Mide alguna orden con ambas alternativas 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 la función gettimeofday, ejecutada justo antes y después de invocar la orden. El tiempo de CPU en modo usuario y sistema se obtiene con la llamada wait3, que devuelve la estructura rusage cuando el proceso termina. Esta estructura se describe en la página de manual de getrusage y está definida en /usr/include/linux/resource.h.

Prueba la opción -v del programa time.

**Entrega:** Compara la información proporcionada por time -v con la de la estructura rusage.

Mide el tiempo 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=1M

$ dd if=/dev/zero of=/var/tmp/prueba oflag=direct count=100K

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

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*).

# ps (~10 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 (~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 en el Campus Virtual), añadiendo la opción -lm para enlazar 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 oom-killer (*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 llevándolas al espacio de intercambio (*swap*).

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.