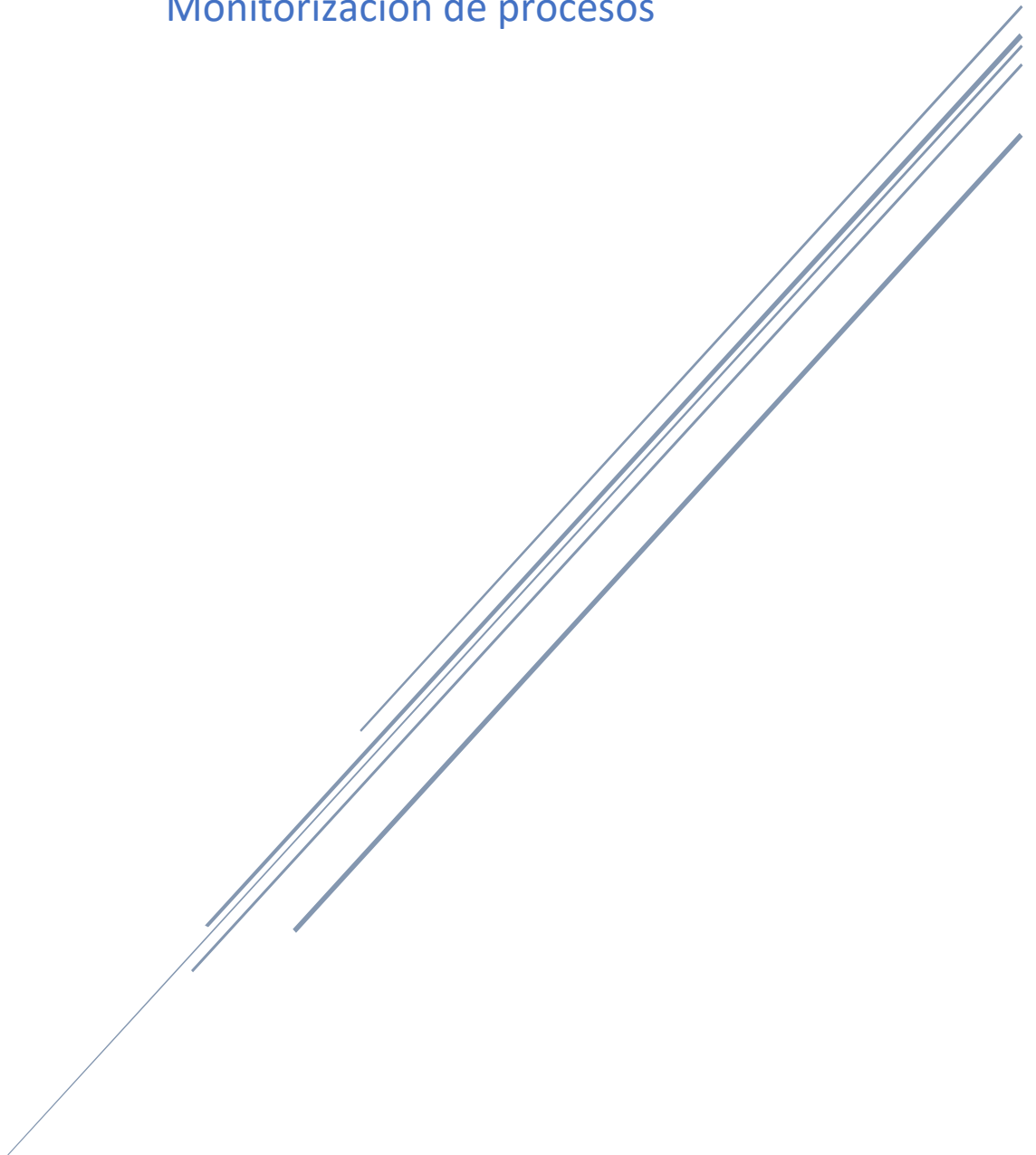


# EVALUACIÓN DE CONFIGURACIONES

Práctica 0

—

Monitorización de procesos



FDI - UCM

Iván Aguilera Calle - Daniel García Moreno

## 1. Información obtenida por time

En este apartado hemos utilizado el comando `time` (y también el programa `time`) para obtener los distintos tiempos de ejecución de distintas órdenes.

Con el programa `time` y utilizando la opción `-v` obtenemos distinta información como el nombre del comando ejecutado, el tiempo de usuario, el porcentaje de uso de la CPU:

```
usuario@debian:/proc$ /usr/bin/time -v xeyes
```

```
Command being timed: "xeyes"
```

```
User time (seconds): 0.04
```

```
System time (seconds): 0.00
```

```
Percent of CPU this job got: 1%
```

```
Elapsed (wall clock) time (h:mm:ss or m:ss): 0:03.67
```

```
...
```

```
usuario@debian:~$ /usr/bin/time -p xeyes
```

```
real 3.37
```

```
user 0.00
```

```
sys 0.03
```

```
usuario@debian:~$ time -p xeyes
```

```
real 5.01
```

```
user 0.04
```

```
sys 0.02
```

Con la opción `-p` el programa `time` muestra los mismos resultados que utilizando la orden: tiempo real (tiempo que realmente ha tardado en ejecutarse), el tiempo de usuario (tiempo empleado por parte del procesador para ejecutar el programa) y el tiempo del sistema (tiempo de las llamadas realizadas al sistema operativo al ejecutar el programa).

## 2. Determinar si las tareas están limitadas por CPU O E/S

En este apartado determinaremos si unos determinados comandos están limitados utilizando para ello la orden `time` con el parámetro `-p`.

- Cuando la cache se encuentra absolutamente vacía requiere de cierto tiempo para la E/S (**IO-bound**):

```
usuario@debian:~/Escritorio$ time -p find /usr > /dev/null  
real 1.37  
user 0.00  
sys 0.14
```

- En cambio, cuando ya no está vacía ya no utiliza tiempo para la E/S:

```
usuario@debian:~/Escritorio$ time -p find /usr > /dev/null  
real 0.02  
user 0.00  
sys 0.00
```

- En la orden ejecutada, tiene mayor tiempo de E/S que, para la CPU (**IO-bound**):

```
usuario@debian:~/Escritorio$ time -p dd if=/dev/zero of=/var/tmp/prueba  
count=1M  
1048576+0 registros leídos  
1048576+0 registros escritos  
536870912 bytes (537 MB) copiados, 1,61107 s, 333 MB/s  
real 1.64  
user 0.10  
sys 0.81
```

- Gasta tiempo mayoritariamente para E/S (**IO-bound**):

```
usuario@debian:~/Escritorio$ time -p dd if=/dev/zero of=/var/tmp/prueba  
oflag=direct count=100K
```

```
102400+0 registros leídos
```

```
102400+0 registros escritos
```

```
52428800 bytes (52 MB) copiados, 5,59777 s, 9,4 MB/s
```

```
real 5.63
```

```
user 0.00
```

```
sys 1.69
```

- Mayor parte del tiempo para E/S (**IO-bound**):

```
usuario@debian:~/Escritorio$ time -p dd if=/dev/urandom  
of=/var/tmp/prueba count=100K
```

```
102400+0 registros leídos
```

```
102400+0 registros escritos
```

```
52428800 bytes (52 MB) copiados, 2,86395 s, 18,3 MB/s
```

```
real 2.86
```

```
user 0.00
```

```
sys 2.69
```

### 3. Comando ps

Hemos escrito un único comando que muestra el usuario, la prioridad, el porcentaje de uso de la CPU y el tamaño de la memoria virtual y física, ordenando de mayor a menor consumo de memoria física.

```
ps -U root -o uname,pri,pcpu,vsz,rss --sort -rss
```

USER	PRI	%CPU	VSZ	RSS
------	-----	------	-----	-----

root	19	0.3	134564	31784
------	----	-----	--------	-------

root	19	0.0	62036	3856
------	----	-----	-------	------

root	19	0.0	75916	3732
------	----	-----	-------	------

root	19	0.0	55908	3576
------	----	-----	-------	------

root	19	0.0	57392	3236
------	----	-----	-------	------

root	19	0.0	9964	2412
------	----	-----	------	------

root	19	0.0	21792	1768
------	----	-----	-------	------

root	19	0.0	52776	1516
------	----	-----	-------	------

...

### 4. Comando top

Al comienzo, el proceso kswapd0 tenía un consumo de un 0% de CPU y tras haber ejecutado el programa “cpu\_mem”, el porcentaje de uso de CPU de cpu\_mem va aumentando, así como el uso de memoria.

Cuando el porcentaje de uso de memoria del proceso “cpu\_mem” es elevado, entra en juego el proceso kswapd0, que va subiendo posiciones en la lista.

Cuando el porcentaje de uso de memoria de “cpu\_mem” es cercano al 100% el proceso “cpu\_mem” finaliza con un out of memory (killed).