

Linux en la enseñanza de Sistemas Operativos

Patricia Paderewski Rodríguez
José Antonio Hernández Gómez
Alejandro León Salas



Índice

- La asignatura en la titulación
- La enseñanza-aprendizaje de los SO
- El concepto de memoria virtual (mv)
- Comprensión del concepto de mv usando Linux
- Conclusiones

La asignatura en la titulación

- Grado en Ingeniería Informática, Doble Grado en Informática y Matemáticas y Doble Grado en Informática y Administración y Dirección de Empresas
- 6 Créditos ECTS: 2 horas de Teoría y 2 horas de Prácticas a la semana en el aula
- Curso: 2º Primer semestre
- Asignatura complementaria: Fundamentos del Software (dada en 1º, primer semestre)
- Dpto. Lenguajes y Sistemas Informáticos

La asignatura en la titulación

| | |
|--------|------------------------------------|
| Tema 1 | Estructuras de sistemas operativos |
| Tema 2 | Procesos e Hilos |
| Tema 3 | Gestión de memoria |
| Tema 4 | Gestión de archivos |
| Tema 5 | Gestión de entradas y salidas |
| Tema 6 | Mecanismos de seguridad |

Índice

- La asignatura en la titulación
- La enseñanza-aprendizaje de los SO
- El concepto de memoria virtual (mv)
- Comprensión del concepto de mv usando Linux
- Conclusiones

La enseñanza-aprendizaje de los SO

- Problemas detectados:
 - Dificultad en la comprensión de los conceptos teóricos básicos.
 - Debido al uso generalizado de las interfaces gráficas para acceder a los servicios del sistema operativo, las abstracciones usadas por éste son cada vez menos visibles.
- ¿Cómo se pueden solucionar?
 - Usar un SO real --> Linux

¿Por qué Linux?

- Licencia *open source* → Se puede acceder sin restricciones al código fuente del kernel y de las herramientas del sistema (compiladores, editores, etc.).
- Es un sistema real que cada vez tiene más usuarios.
- Tiene la ventaja de que se puede desacoplar la interfaz gráfica del SO en sí → esto facilita que los estudiantes capten las abstracciones que proporciona el sistema: Procesos, Archivos, etc.
- Hay mucha información sobre el diseño e implementación de Linux que los estudiantes pueden consultar

¿Cómo lo hacemos con Linux?

- Desde este enfoque, algunos de los aspectos más destacables en el estudio de Linux son:
 - *Implementación de procesos e hilos*. Linux implementa el concepto de hilo y nos permite explicar la diferencia usando las estructuras internas que ellos pueden visualizar.
 - *Planificador de procesos*. Les explicamos el problema de planificación de procesos y, en particular, el de Linux. En prácticas pueden comprobar estos conceptos e incluso cambiar el orden de ejecución de los procesos actuando sobre su prioridad con las herramientas proporcionadas por el sistema operativo.
 - *Gestión de memoria*. La memoria es fundamental. Se puede ver cómo Linux realiza esta función.

¿Por qué Linux?

- *Programación de la API (Application Programming Interface)*. El uso de las llamadas al sistema hace que el estudiante afiance conceptos de los SO.
- *Administración del sistema*. El uso de herramientas de monitorización y administración hace que se tenga una visión global del sistema
- *Configuración y compilación del kernel*. Es importante que los estudiantes sepan que al final, el SO es un sistema software que se puede ampliar, mejorar y hacerlo evolucionar para adaptarse a nuevas necesidades.
- *Implementación de módulos de carga (LKM)*. Los módulos de carga dinámica permiten al estudiante modificar la funcionalidad del sistema en tiempo de ejecución y entender mejor el funcionamiento del mismo.
- *Acceso a datos del sistema*. El uso del pseudo-sistema de archivos */proc* permite ilustrar algunos aspectos de funcionamiento del sistema gracias al acceso en tiempo real a los datos del kernel.

Índice

- La asignatura en la titulación
- La enseñanza-aprendizaje de los SO
- El concepto de memoria virtual (mv)
- Comprensión del concepto de mv usando Linux
- Conclusiones

¿Qué es la memoria virtual?

- Para ejecutar un programa es necesario cargarlo en memoria principal (RAM).
- La Memoria Virtual (MV) permite:
 - Ejecutar un programa con un tamaño superior a la memoria RAM del ordenador.
 - Ejecutar muchos programas cuya suma de tamaño es superior a la RAM del ordenador.

¿Qué es la memoria virtual?

- Elementos que permiten que la mv funcione:
 - Principio de localidad de los programas.
 - El espacio de direcciones del programa es lógico, no se corresponde con el espacio físico de direcciones de la RAM.
 - Conjunto residente.

¿Qué es la memoria virtual?

- Espacio de direcciones virtual.
- Cada proceso tiene un espacio de direcciones virtual único.
- El espacio virtual se divide en trozos: segmentos (regiones) o páginas.
- Un proceso no tiene acceso a todas las páginas de su espacio de direcciones virtual, solamente las que tienen contenido: válidas frente a no válidas.
- *Memory Management Unit* (MMU)

¿Qué es la memoria virtual?

- Si una página válida reside en almacenamiento secundario, el proceso no puede referenciarla hasta que sea cargada en memoria RAM.
- Si el proceso referencia la página entonces MMU genera un código de error: falta de página.
- El kernel interviene para traer la página desde almacenamiento secundario a RAM.

¿Qué es la memoria virtual?

- ¿Es necesario retirar páginas de la RAM?
 - Si crece mucho el número de procesos, puede que no quepan todos sus conjuntos residentes y habrá que retirar páginas y almacenarlas en disco.
 - Si no hay espacio suficiente para incorporar páginas como resultado de una falta de página habrá que retirar páginas que en el momento actual no son necesarias.

¿Qué es la memoria virtual?

- Compartición de páginas de RAM entre espacios de direcciones virtuales de distintos procesos. Ejemplos:
 - Compartir la biblioteca de C
 - Compartir el código de una aplicación tipo procesador de textos.

¿Qué es la memoria virtual?

- Regiones. Grupo de páginas identificable que permite asociar propiedades al conjunto.
- Regiones por defecto para cualquier proceso:
 - Región de texto (código).
 - Región de datos.
 - Región de pila (*stack*).
 - Región de *heap*.

Índice

- La asignatura en la titulación
- La enseñanza-aprendizaje de los SO
- El concepto de memoria virtual (mv)
- **Comprensión del concepto de mv usando Linux**
- Conclusiones

¿Cómo implementa Linux la mv?

- Gestión de memoria mediante regiones y páginas (segmentación paginada)
- Uso de memoria de almacenamiento secundario para extensión de la memoria RAM:
 - Partición en disco para intercambio.
 - Archivo de intercambio en el sistema de archivos.

Segmentación paginada

- Hay que implementar el concepto de region en el espacio virtual para descartar las áreas no válidas.
- ¿Cómo se representa el espacio de direcciones virtual de un proceso? mm_struct

```
struct mm_struct {  
    ...  
    unsigned long task_size; /* size of task vm space */  
    ...  
    unsigned long start_code, end_code, start_data, end_data;  
    unsigned long start_brk, brk, start_stack;  
    unsigned long arg_start, arg_end, env_start, env_end;  
    ...  
}
```

Segmentación paginada

- ¿Y el resto de regiones posibles del espacio de direcciones? `mm_struct` incluye una lista de `vm-areas`.

```
struct mm_struct {  
    ...  
    unsigned long task_size; /* size of task vm space */  
    ...  
    unsigned long start_code, end_code, start_data, end_data;  
    unsigned long start_brk, brk, start_stack;  
    unsigned long arg_start, arg_end, env_start, env_end;  
    ...  
    struct vm_area_struct * mmap; /* list of VMAs */  
}
```

Segmentación paginada

- ¿Y qué información contiene una vm-area?

vm_area_struct

```
struct vm_area_struct {  
    ...  
    struct mm_struct* vm_mm;    /* The address space we belong to. */  
    unsigned long vm_start;    /* Our start address within vm_mm. */  
    unsigned long vm_end;    /* The first byte after our end address  
                               within vm_mm. */  
  
    pgprot_t vm_page_prot;    /* Access permissions of this VMA. */  
  
    /* Information about our backing store: */  
    unsigned long vm_pgoff;    /* Offset (within vm_file) in PAGE_SIZE  
                               units, *not* PAGE_CACHE_SIZE */  
  
    struct file* vm_file;    /* File we map to (can be NULL). */  
    void* vm_private_data;    /* was vm_pte (shared mem) */  
};
```

Espacio de intercambio

- Partición de intercambio dedicada. Se crea igual que una partición normal pero está etiquetada como swap. Ejemplo /dev/swap1
- Archivo de intercambio. Ejemplo ~/swapFile
- ¿Cómo se determina que es de intercambio?

```
#> mkswap /dev/swap1
```

```
#> mkswap /swapFile
```

Espacio de intercambio

- ¿Cómo hacer que el kernel use un espacio de intercambio previamente creado?

```
#> swapon /dev/swap1
```

```
#> swapon /swapFile
```


Linux para facilitar la comprensión del concepto mv

- Podemos utilizar tres niveles de acercamiento:
 - Órdenes del shell
 - Sistema de archivos virtual /proc
 - Herramientas gráficas

Órdenes del shell

- free:

\$> free -wh

| | total | used | free | shared | buffers | cache | available |
|-------|-------|------|------|--------|---------|-------|-----------|
| Mem: | 11G | 2,5G | 6,3G | 154M | 352M | 2,2G | 8,4G |
| Swap: | 11G | 0B | 11G | | | | |

Órdenes del shell

- top:

```
aleon@aleon-ThinkPad-W540: ~
top - 09:34:35 up 2:31, 1 user, load average: 0,69, 0,29, 0,21
Tasks: 284 total, 2 running, 218 sleeping, 0 stopped, 0 zombie
%Cpu(s): 1,2 us, 0,5 sy, 0,0 ni, 98,1 id, 0,1 wa, 0,0 hi, 0,0 si, 0,0 st
KiB Mem : 11917568 total, 6585416 free, 2646400 used, 2685752 buff/cache
KiB Swap: 12198908 total, 12198908 free, 0 used. 8718504 avail Mem
```

| PID | USER | PR | NI | VIRT | RES | SHR | S | %CPU | %MEM | TIME+ | COMMAND |
|------|-------|-----|----|---------|--------|--------|---|------|------|---------|-----------------|
| 3603 | aleon | 20 | 0 | 1910504 | 308568 | 130280 | S | 6,6 | 2,6 | 2:23.47 | Web Content |
| 1170 | root | 20 | 0 | 463792 | 139344 | 106528 | S | 2,0 | 1,2 | 3:30.34 | Xorg |
| 5534 | aleon | 20 | 0 | 2360036 | 303200 | 119932 | S | 2,0 | 2,5 | 1:05.86 | thunderbird |
| 4321 | aleon | 20 | 0 | 1389200 | 159664 | 97656 | S | 0,7 | 1,3 | 0:30.08 | chromium-browse |
| 8 | root | 20 | 0 | 0 | 0 | 0 | I | 0,3 | 0,0 | 0:01.50 | rcu_sched |
| 1187 | root | -51 | 0 | 0 | 0 | 0 | S | 0,3 | 0,0 | 1:15.46 | irq/32-nvidia |
| 2424 | aleon | 20 | 0 | 1436352 | 179568 | 91056 | S | 0,3 | 1,5 | 2:18.06 | compiz |
| 3183 | aleon | 20 | 0 | 671196 | 41472 | 30852 | S | 0,3 | 0,3 | 0:09.26 | gnome-terminal- |
| 3802 | aleon | 20 | 0 | 1805292 | 230808 | 96236 | S | 0,3 | 1,9 | 0:12.11 | Web Content |
| 4348 | aleon | 20 | 0 | 1280856 | 103096 | 80032 | S | 0,3 | 0,9 | 0:00.29 | chromium-browse |
| 4508 | aleon | 20 | 0 | 1814360 | 142444 | 81012 | S | 0,3 | 1,2 | 0:13.36 | chromium-browse |
| 5886 | root | 20 | 0 | 0 | 0 | 0 | I | 0,3 | 0,0 | 0:00.78 | kworker/0:2 |
| 6879 | aleon | 20 | 0 | 43172 | 4068 | 3312 | R | 0,3 | 0,0 | 0:00.50 | top |

Órdenes del shell

- vmstat:

```
aleon@aleon-ThinkPad-W540: ~  
aleon@aleon-ThinkPad-W540:~$ vmstat  
procs  -----memory-----  ---swap--  -----io----  -system--  -----cpu-----  
r  b    swpd    free    buff    cache    si    so    bi    bo    in    cs  us  sy  id  wa  st  
0  0        0 6573044 365900 2342616    0    0    24    12    56   344  2   1  97   0   0  
aleon@aleon-ThinkPad-W540:~$
```

Sistema de archivos /proc

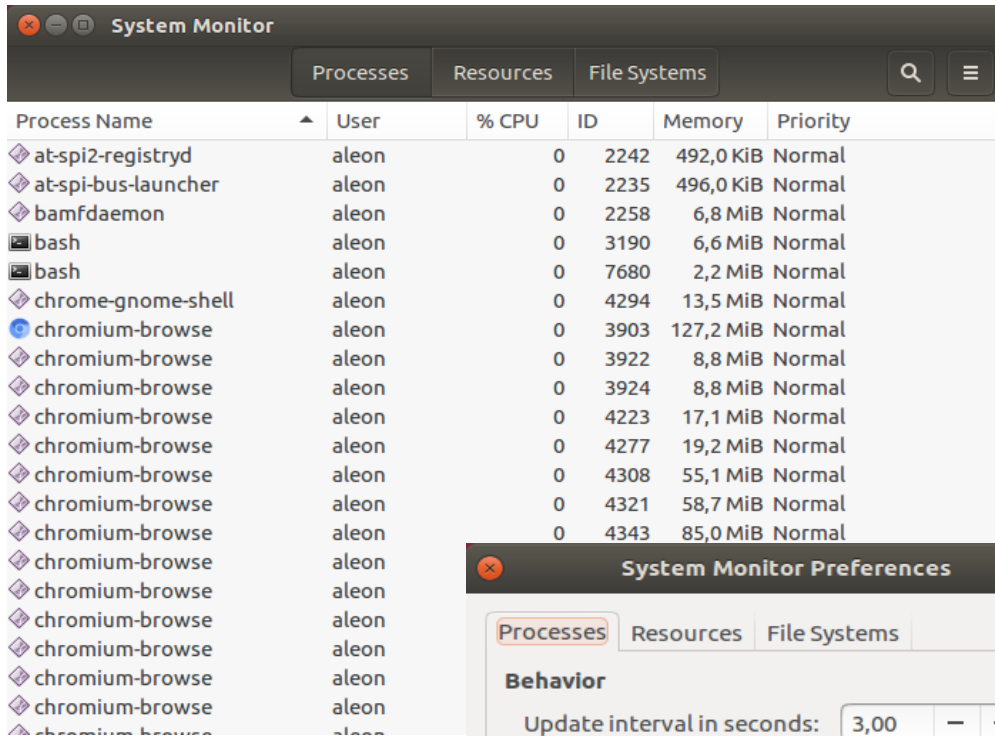
- [/proc/meminfo](#): muestra la información relativa al uso de memoria del sistema
 - La información es mucho más amplia que la proporcionada por las ordenes del shell
- [/proc/<pid>/maps](#): contiene las regiones del espacio de memoria virtual que son válidas de un proceso.

Sistema de archivos /proc

- /proc/<pid>/maps: ejemplo

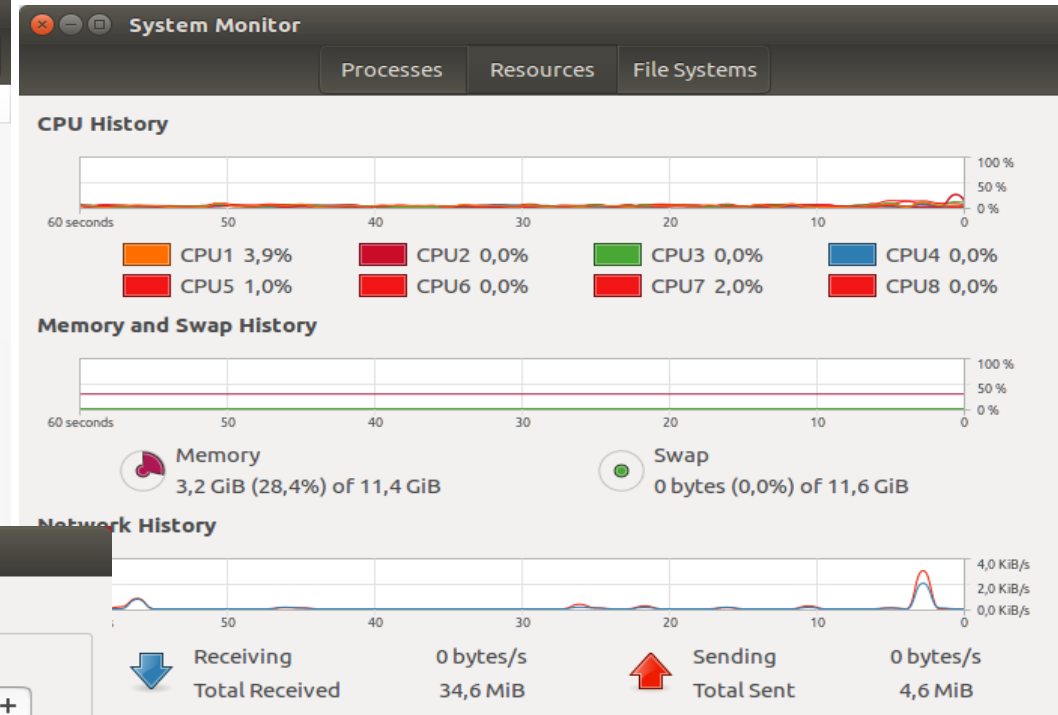
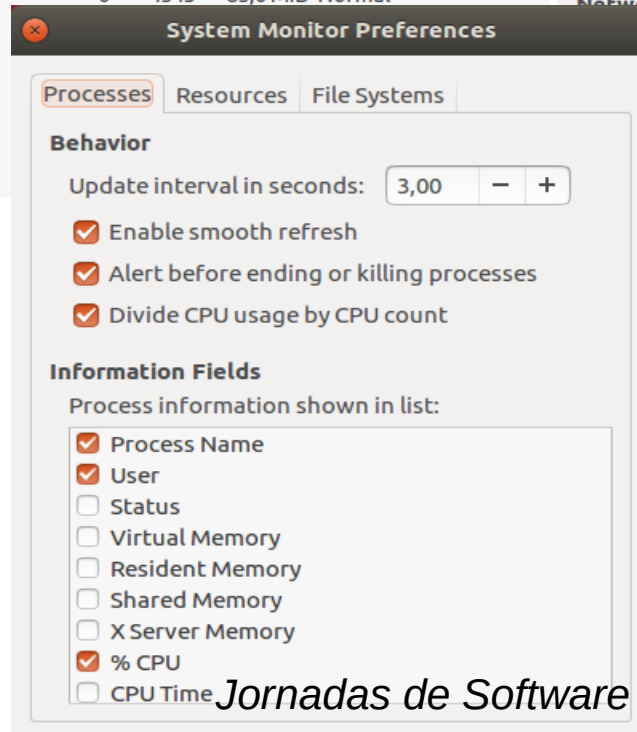
| address | perms | offset | dev | inode | pathname |
|---------------------------|-------|----------|-------|----------|---|
| 00400000-00401000 | r-xp | 00000000 | 08:05 | 15338259 | /home/aleon/src/displayVirtualAddresses |
| 00600000-00601000 | r--p | 00000000 | 08:05 | 15338259 | /home/aleon/src/displayVirtualAddresses |
| 00601000-00602000 | rw-p | 00001000 | 08:05 | 15338259 | /home/aleon/src/displayVirtualAddresses |
| 01efe000-01f1f000 | rw-p | 00000000 | 00:00 | 0 | [heap] |
| 7fe950c74000-7fe950e34000 | r-xp | 00000000 | 08:05 | 15073540 | /lib/x86_64-linux-gnu/libc-2.23.so |
| 7fe950e34000-7fe951034000 | ---p | 001c0000 | 08:05 | 15073540 | /lib/x86_64-linux-gnu/libc-2.23.so |
| | | | | | |

GNOME system monitor

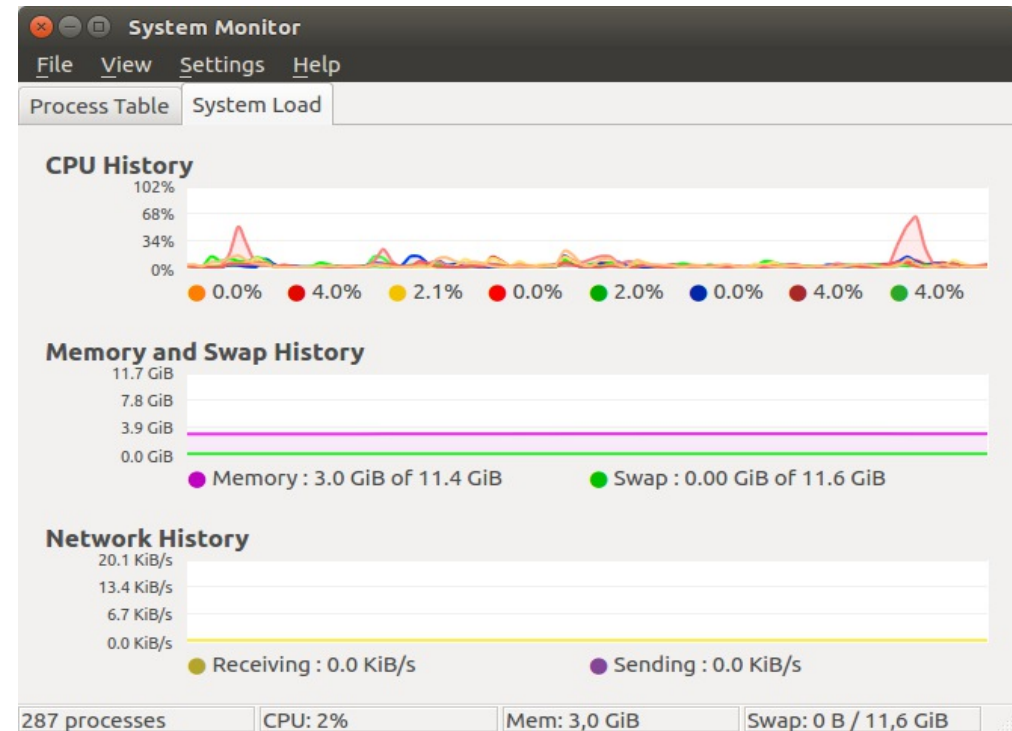
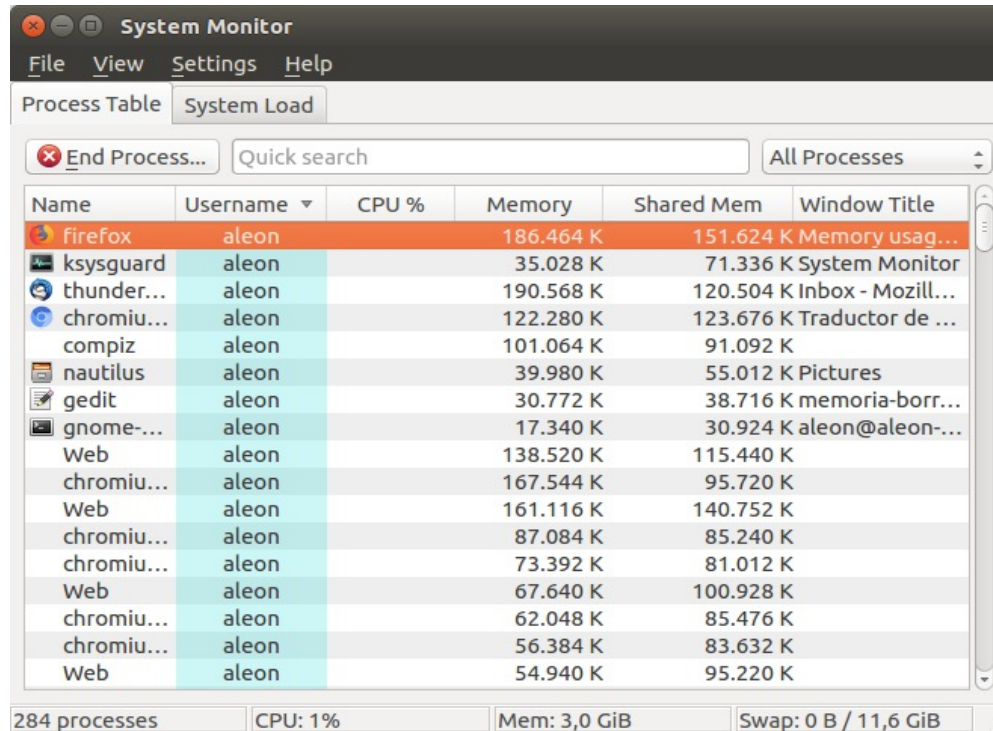


The screenshot shows the 'Processes' tab of the GNOME System Monitor. It displays a list of running processes with columns for Process Name, User, % CPU, ID, Memory, and Priority. The list includes system processes like 'at-spi2-registryd' and user processes like 'bash' and multiple instances of 'chromium-browser'.

| Process Name | User | % CPU | ID | Memory | Priority |
|---------------------|-------|-------|------|-----------|----------|
| at-spi2-registryd | aleon | 0 | 2242 | 492,0 KiB | Normal |
| at-spi-bus-launcher | aleon | 0 | 2235 | 496,0 KiB | Normal |
| bamfd daemon | aleon | 0 | 2258 | 6,8 MiB | Normal |
| bash | aleon | 0 | 3190 | 6,6 MiB | Normal |
| bash | aleon | 0 | 7680 | 2,2 MiB | Normal |
| chrome-gnome-shell | aleon | 0 | 4294 | 13,5 MiB | Normal |
| chromium-browser | aleon | 0 | 3903 | 127,2 MiB | Normal |
| chromium-browser | aleon | 0 | 3922 | 8,8 MiB | Normal |
| chromium-browser | aleon | 0 | 3924 | 8,8 MiB | Normal |
| chromium-browser | aleon | 0 | 4223 | 17,1 MiB | Normal |
| chromium-browser | aleon | 0 | 4277 | 19,2 MiB | Normal |
| chromium-browser | aleon | 0 | 4308 | 55,1 MiB | Normal |
| chromium-browser | aleon | 0 | 4321 | 58,7 MiB | Normal |
| chromium-browser | aleon | 0 | 4343 | 85,0 MiB | Normal |



Ksysguard (KDE)



Conclusiones

- Cualquier concepto de SO, por simple que pueda parecer, conlleva muchos detalles.
- La posibilidad de estudiar código fuente del sistema operativo permite mostrar claramente la implementación de conceptos.
- Las órdenes del shell permiten de una forma más asequible comprobar información del sistema.
- El /proc permite comprender de forma más precisa las estructuras del kernel de Linux.
- Las herramientas gráficas son fáciles de usar y aprender pero dan muy poca información.

Gracias por vuestra atención
:-) :-)

*Jornadas de Software Libre de la UGR,
27 y 28 de septiembre de 2018*