SISTEMAS OPERATIVOS I

Práctica 1: El entorno Linux y el intérprete de comandos o *shell*

Objetivo

- Familiarizarse con los entornos de Linux.
- Obtener información sobre la organización del sistema de ficheros y de los procesos.
- Familiarizarse con los comandos *ps, top, pstree, ls, cd, pwd, cd, cp, rm, rmdir, mkdir, more, less, cat, wc, diff, mv, grep, sort, date* del intérprete de comandos.

Información útil para la práctica

Arranca el ordenador con el SO Linux (distribución Ubuntu con entorno gráfico Gnome). Utiliza el comando *man nombre_de_comando* para comprobar la funcionalidad de los comandos o busca en internet dicha funcionalidad. Es conveniente que encuentres una fuente de consulta en Internet sobre órdenes del intérprete de comandos que sea útil como referencia de trabajo futuro.

Comandos para obtener información sobre los procesos

1. Lanza un terminal. Ejecuta los comandos *ps, top*. Leer el man de estos comandos. Prueba las opciones *-a, -e, -f, -x, -u* del comando *ps*. Estas opciones permiten cambiar los procesos que se muestran y la información sobre los mismos; consulta el *man* para ver la información que se muestra en cada caso. Puedes usar varias opciones al mismo tiempo, por ejemplo, *ps -ef*.

2. Con las opciones apropiadas, obtén el árbol de procesos usando los comandos *ps* (identifica el usuario propietario de cada proceso y el padre del proceso) y *pstree*. Compáralo con el obtenido con el interfaz gráfico.

3. Lee el mapa de memoria de un proceso en el fichero */proc/PID/maps*, donde PID es el identificador de uno de los procesos, utilizando los comandos *more* y/o *cat*. Identifica las direcciones de memoria de comienzo y final de cada módulo y los permisos. Consulta el uso del comando *pmap*.

Comandos para el sistema de archivos

1. Analiza la jerarquía de directorios con los comandos *ls* y *cd*. Utiliza las opciones *-a* y *-l* del *ls*. Compárala con la obtenida usando el interfaz gráfico. ¿Qué significan las entradas "." y ".." que se encuentran en cualquier directorio?. Comprueba el funcionamiento de las siguientes órdenes: *cd, cd ., cd ..*

2. Pon un ejemplo de uso de los comandos anteriores con una ruta absoluta (comenzando en el directorio raíz) y otra relativo (comenzando en el directorio actual). Utiliza el comando *pwd* para comprobar cual es el directorio de trabajo.

3. Comprueba la funcionalidad de otros comandos relacionados con el sistema de archivos, por ejemplo, *cp, rm* (opciones *-i, -r*, ¡cuidado con la opción -r !), *rmdir*, *mkdir*.

Otros comandos y utilización de comodines

1. Utiliza el comando *more* para visualizar archivos de texto en el terminal. Comprueba que ocurre al pulsar la barra espaciadora, ENTER o la letra *q* durante la visualización de un archivo que no se ve completo en el monitor. Comprueba la funcionalidad del comando *less*.

2. Comprueba la funcionalidad de otros comandos, por ejemplo: *cat, wc, diff, mv, grep, sort, date*.

3. Emplea los comodines (*wildcard*): *, ?, [ ] y { } con alguno de los comandos que has utilizado en los ejercicios anteriores, por ejemplo, para listar algunos archivos del directorio */dev*. Piensa en algún otro ejemplo.

4. Usa las redirecciones (>, >>, <, <<) y tuberías (|) junto con algunos comandos vistos en ejercicios anteriores, centrándote en las redirecciones de la salida estándar y las tuberías. Es más difícil encontrar ejemplos útiles de redirecciones de la entrada estándar. Puedes consultar el siguiente enlace http://hipertextual.com/archivo/2014/07/redirecciones-y-tuberias-bash/.

Utilización en programas en C

1. Haz un programa en C que consuma mucha memoria. Debes ajustar el programa para que ocupe la memoria de manera progresiva. El programa se basará en un lazo con las función *sleep* y *malloc* y a cada iteración se reservará y utilizará más memoria (escribiendo datos). Comprueba como se va agotando la memoria física disponible.

2. Haz un programa en C que consuma mucha CPU. Para ello programa un lazo de muchas iteraciones de modo que en cada iteración se realicen una o varias operaciones sobre operandos tipo *float* o *double*. Analiza el porcentaje de uso de CPU con la monitorización del sistema.

3. Comprueba como es el programa ensamblador correspondiente al código C anterior. Para ello debes usar una determinada opción del compilador *gcc* que permite obtener un fichero con el código en ensamblador (leer el man para obtener la opción).