



INSTRUCCIONS IMPORTANTS

| | |
|-----------|----------------------------|
| 🕒 Created | @January 15, 2021 10:31 AM |
| 📅 Date | |
| 🏷️ Tags | |
| 🏷️ Tema | |

Trabajo con ficheros



grep → Busca texto (o patrones de texto) en ficheros.



chmod → Modifica los permisos de acceso a un fichero.



env → Ejecuta un comando en un entorno modificado, si no se le pasa comando, muestra el entorno.

Utilities



sprint → Conversión de formato almacenándola en un búffer.



atoi → Convierte un string a un número entero.



indent → Indentación de ficheros fuente.

Comandos útiles para trabajar con Signals



getpid → Retorna el PID del proceso que la ejecuta.



fork → Crea el proceso nuevo, hijo del que la ejecuta



exit → Termina el proceso que ejecuta la llamada.



waitpid → Espera la finalización de un proceso hijo.



execlp → Ejecuta un programa en el contexto del mismo proceso.



perror → Escribe un mensaje del último error producido.



ps → Devuelve información de los procesos.



proc → Pseudo-file system que ofrece información de datos del kernel.

Programacion signals



`sigaction` → Reprograma la acción asociada a un evento concreto.



`kill` → Envía un evento concreto a un proceso.



`sigsuspend` → Bloquea el proceso que la ejecuta hasta que recibe un signal (los signals cuyo tratamiento es ser ignorado no desbloquean el proceso).



`sigprocmask` → La instrucción **sigprocmask** se usa para bloquear o desbloquear los signals que tenemos guardados en la máscara. El comportamiento exacto de la función depende del primer parámetro, que puede tomar los siguientes valores:

- **SIG_BLOCK**: El conjunto de signals de la máscara se añadirán al conjunto de signals bloqueados en el programa actual.
- **SIG_UNBLOCK**: El conjunto de signals de la máscara se eliminarán del conjunto de signals bloqueados en el programa actual.
- **SIG_SETMASK**: El conjunto de signals bloqueados en el programa actual, pasará a ser la máscara.

Reservas de memoria



`gcc -static` → En sistemas que soporten l'enllaç dinàmic, impedeix l'enllaç de llibreries compartides. En canvi, en els sistemes que no soporten aquest tipus d'enllaç not té cap efecte.



nm → La crida a sistema nm imprimeix la llista de tots els símbols dels arxius objecte. Si no escrivim cap fitxer objecte com a argument de la crida, el sistema operatiu assumeix l'arxiu "a.out". Cada símbol que retorna aquesta crida té un significat diferent.



objdump → Imprimeix la informació sobre un o més arxius objecte. Les opcions controlen quina informació volem mostrar. Aquesta informació és útil pels programadors que treballen desenvolupant els compiladors.

objdump -d → Imprimeix les instruccions mnemotècniques en llenguatge ensamblador del arxiu objecte. Aquesta instrucció només desansembla aquelles seccions que poden contenir instruccions.



/proc → El sistema d'arxius proc és un pseudo sistema d'arxius que provee una interfície al kernel de les diferents estructures de dades. Normalment s'emmagatzema a la carpeta /proc. S'acostuma a construir automàticament pel sistema, però també es pot generar de forma manual utilitzant la instrucció: `mount -t proc proc /proc`

/proc/[pid]/maps → Un fitxer que conté les regions de memòria mapades actualment i els seus permisos d'accés. L'arxiu que retorna conté les següents columnes: address, perms, offset, dev, inode i pathname. El camp d'adreça és l'espai d'adreces en el procés que el fitxer ocupa en memòria. La columna perms conté els diferents permisos (read, write, execute, shared, private). La columna offset és el offset en l'arxiu. I el pathname normalment és l'arxiu que està copiant el mapejat.



malloc → Coloca els n bytes i retorna un punter a la memòria. La memòria no s'inicialitza. Si la mida és 0, llavors la funció ens retorna un punter NULL, o el valor d'un punter únic que després passarà a lliure.



free → Allibera l'espai de memòria apuntat per ptr. Si el punter apunta a un lloc NULL, no es realitza cap operació.



sbrk → S'utilitza per canviar dinàmicament la quantitat d'espai assignat per al segment de dades del procés de trucada. El canvi es realitza restablint el valor de ruptura del procés i assignant la quantitat d'espai adequat.

Memoria en ejecucion



nice → Ejecuta un programa modificándole la prioridad de planificación.



uptime → Muestra cuánto tiempo lleva encendido el sistema y la carga media.



w → Muestra quien está conectado y que está haciendo.



time → Ejecuta un programa y mide el tiempo que tarda en ejecutarse.



vmstat → Muestra estadísticas sobre el uso de la memoria.



top → Muestra información sobre el sistema y los procesos en ejecución.

Drivers y modulos Kernel



mknod → Comando que crea un fichero especial.



insmod → Comando que inserta un módulo en el kernel.



rmmod → Comando que descarga un módulo del kernel.



lsmod → Comando que muestra el estado de los módulos cargados en el kernel.



sudo → Comando que permite ejecutar un comando como root.



open → Abre un fichero o dispositivo.



write → Llamada a sistema para escribir en un dispositivo.



read → Llamada a sistema para leer de un dispositivo virtual.



strace → Lista de llamadas a sistema ejecutadas por un proceso.

Pipes y sockets



pipe → Llamada a sistema para crear una pipe sin nombre.



dup/dup2 → Duplica un descriptor de fichero.



socket → Crea un socket.



bind → Asigna un nombre o dirección a un socket.



listen → Espera conexiones a un socket.



accept → Acepta una conexión en un socket.



connect → Inicia una conexión a un socket.

Sistema de ficheros



open/creat → Abre/crea un fichero o dispositivo.



df → Devuelve información sobre el sistema de ficheros.



ln → Crea enlaces (links) a ficheros.



namei → Procesa una ruta de una fichero hasta encontrar el punto final.



readlink → Lee el contenido de un link simbólico.



stat → Muestra información de control de un fichero.



Iseek → Modifica la posición de lectura / escritura de un fichero.