

TODO PROGRAMA EN EJECUCIÓN ES UN PROCESO.

ps sirve para mostrar información sobre procesos ps -t → termlist, -g grouplist, -o format, -u userlist.

<unistd.h> Librería que contiene funciones para el manejo de directorios y archivos.

<sys/types.h> Contiene funciones de búsqueda y ordenamiento de directorios y manipulación de archivos.

-----Jerarquía y estados de un proceso.-----

Un solo proceso sólo tiene un padre y 0 o mas hijos

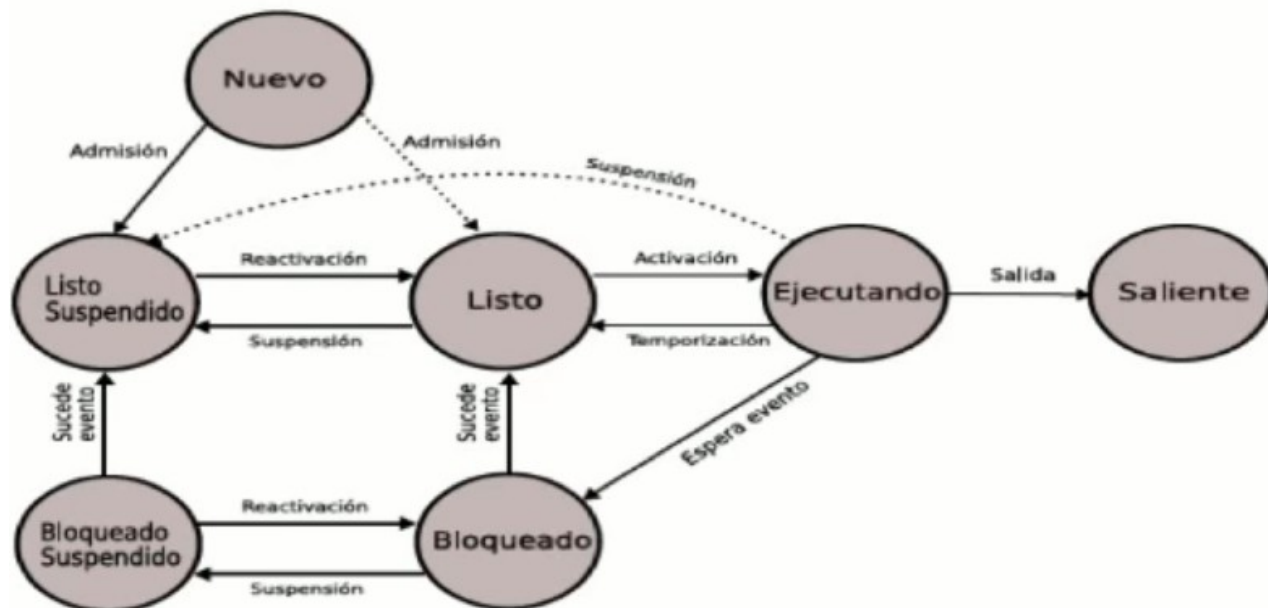
Nuevo → Proceso nuevo entrante en ejecución, no ha sido cargado en memoria. Se pretenden liberar recursos

Listo → Cuando el SO estime que hay memoria y recursos suficientes para ejecutar el proceso, no se ejecutara (no tendrá la CPU) pero tendrá su mapa de memoria asociado en memoria.

Bloqueado → cuando un proceso se está ejecutando en la CPU y llega un momento que necesita acceder al disco, o hay un evento de solicitar algo, o fallo de paginación, en ese momento se bloquea y deja de necesitar la CPU, el SO lleva el proceso a memoria o a disco

Saliente → Termine su estado de ejecución.

Listo suspendido o bloqueado suspendido → Los procesos listos y bloqueado en vez de estar los procesos en memoria esperando si está bloqueado a que suceda determinado evento o si está listo que le den la CPU ahora estarán en Disco Duro, teniendo una menor prioridad al momento de coger la CPU



-----Fork()-----

En linux Fork está implementado usando paginas de copia en escritura (copy-on-write) ai que la unica falta que incurre es en el tiempo y memoria requeridos para duplicar las tablas de páginas del padre y para crear una única estructura de tareas para el hijo.

fork() → devuelve un entero 1 Padre 0 Hijo. Getpid() o getppid() y para obtener el idetificador del usuario con getuid.

-----PCB Process Control Block-----

TABLA DE PROCESO es un ARRAY que lleva información de todos los procesos que se están ejecutando en un sistem a.

-----TERM Signals-----

<signal.h>

signal(“indica numero de la señal”, &“funcion a llamar”)

signal(SIGIO, &funcion) //Se puede ver como si SIGIO fuera una interrupcion y la funcion la rutina de atencion a la interrupcion

void funcion(int sig){} → el entero que se recibe en la función es el numero de la señal

Señal es una notificación que se envia a un proceso determinado

ps u y ver programa que esta con la señal

kill -l → ver las señales que hay

SIGKILL → Terminar un procceso

SIGCONT → continuar un proceso pausado

SIGSTOP → Pausar un proceso

SIGHUP → Cuando se cierra una terminal asociada a procesos, se envia a los mismos señal para que se detengan.

SIGABRT → Para abortar un proceso.

SIGIO →

kill -”señal que esta siendo usada” pid

¿CUANDO NO SE ENVIA UNA FUNCION SIGNAL HANDLER QUÉ SUCEDE?

Action comportamiento por defecto al recibir una señal.

sTERM por defecto terminara el proceso

sIGN por defecto ignora la señal

sCORE Similar a term y genera un archivo llamado dump core

sSTOP si es stop para

sCONT continua

IGNORAR una señal

signal(“señal a la que se cmabiara el comportamiento”, SIG_IGN) Así ignoramos la señal.

La señal SIGKILL Y SIGSTOP no se pueden cambiar sus accione spor defecto