Trabajo Practico Nº 1.

# Alumno: Tomas Alaluf, Gonzalo Errandonea, Gonzalo Romero. Profesor: Osvaldo Aguiar. Instituto: UADER FCYT. Carrera: Licenciatura en sistemas de información. Materia: Sistemas Operativos.

**1) Defina proceso**

Un proceso se lo puede definir como una entidad que consiste en un número de elementos.

Los dos elementos esenciales serían el código de programa (que puede compartirse con otros procesos que estén ejecutando el mismo programa) y un conjunto de datos asociados a dicho código.

Otras definiciones:

Un programa en ejecución.

Una instancia de un programa ejecutado en un computador.

La entidad que se puede asignar y ejecutar en un procesador.

Una unidad de actividad que se caracteriza por la ejecución de una secuencia de instrucciones, un estado actual, y un conjunto de recursos del sistema asociados.

**2) Dibuje un diagrama de estados con los estados principales de un proceso.**



**3) Explique cada uno de los estados en el diagrama solicitado anteriormente.**

• Ejecutando. El proceso está actualmente en ejecución. Para este capítulo asumimos que el computador tiene un único procesador, de forma que sólo un proceso puede estar en este estado en un instante determinado.

• Listo. Un proceso que se prepara para ejecutar cuando tenga oportunidad.

• Bloqueado. Un proceso que no puede ejecutar hasta que se cumpla un evento determinado o se complete una operación E/S.

• Nuevo. Un proceso que se acaba de crear y que aún no ha sido admitido en el grupo de procesos ejecutables por el sistema operativo. Típicamente, se trata de un nuevo proceso que no ha sido cargado en memoria principal, aunque su bloque de control de proceso (BCP) si ha sido creado.

• Saliente. Un proceso que ha sido liberado del grupo de procesos ejecutables por el sistema operativo, debido a que ha sido detenido o que ha sido abortado por alguna razón.

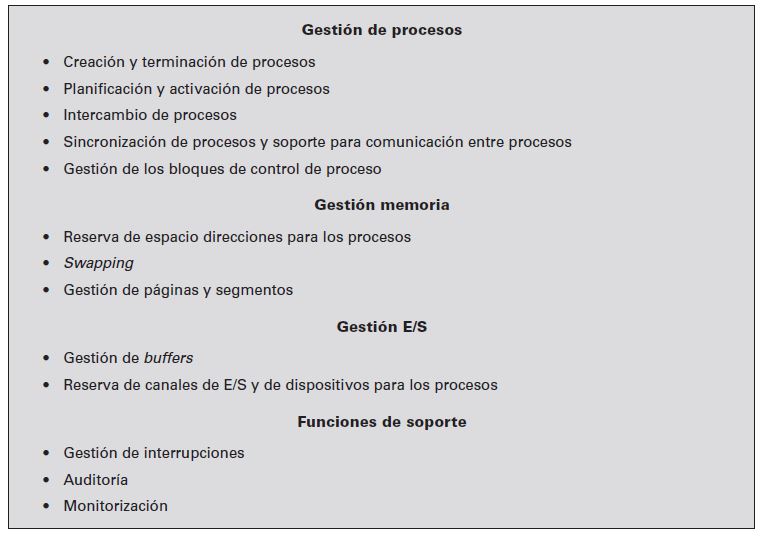
**4) Cuando una Interrupción o una llamada al sistema transfiere el control al sistema operativo, por lo general se utiliza un área de la pila distinta a la pila del proceso. Explique cuál es la razón.**

Existen muchas razones razones, dos de ellas pueden ser las siguientes:

- Para evitar que el SO colapse debido a que un usuario escribió mal un programa y no reservó suficiente espacio en la pila.

- Si el kernel deja datos de la pila en el espacio libre de la memoria de un programa de un usuario en el retorno desde una llamada del sistema, un usuario malintencionado podría usar datos para hallar información acerca de los otros procesos.

**5) ¿Cuales son las actividades principales que el SO realiza en la gestión de procesos?**



**6) Un proceso que no está en memoria principal no está disponible de forma inmediata para su ejecución, independientemente de si está a la espera o no de un evento.**

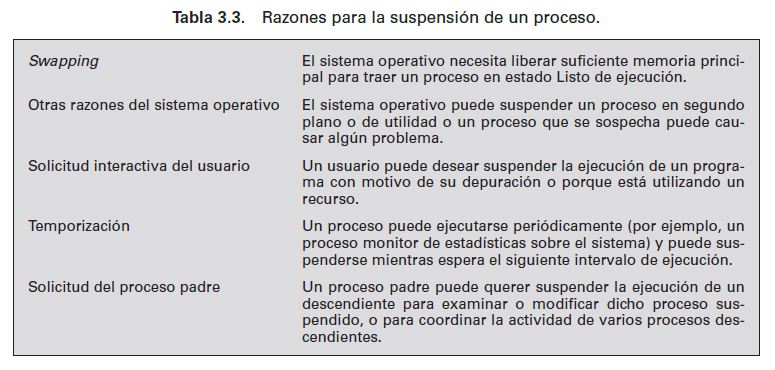
Podemos analizar el concepto de procesos suspendidos, definiendo un proceso suspendido como el que cumple las siguientes características:

1. El proceso no está inmediatamente disponible para su ejecución.

2. El proceso puede estar o no a la espera de un evento, si es así, la condición de bloqueo es independiente de la condición estar suspendido, y si sucede el evento que lo bloquea, eso no habilita al proceso para su ejecución inmediata.

3. El proceso fue puesto en estado suspendido por un agente: bien el proceso mismo, el proceso padre o el sistema operativo, con el propósito de prevenir su ejecución.

4. El proceso no puede ser recuperado de este estado hasta que el agente explícitamente así lo indique.

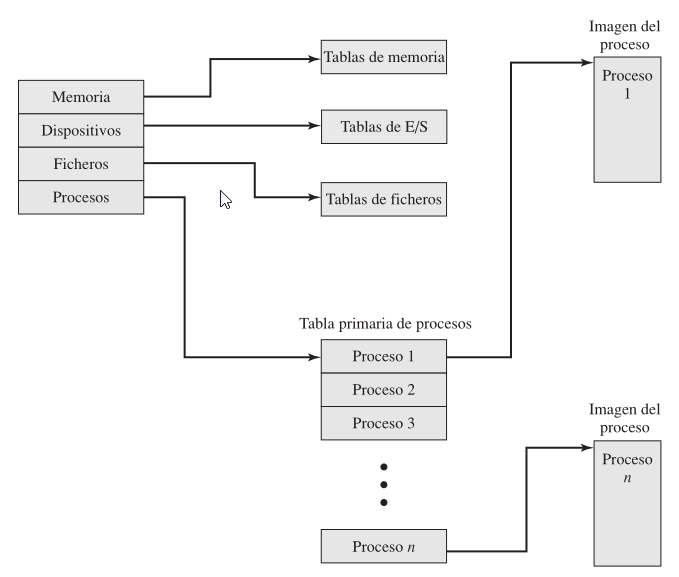


**7) Explique detalladamente que es el intercambio de procesos.**

El Swapping mueve en parte o todo el proceso de memoria principal al disco. Cuando ninguno de los procesos en memoria principal se encuentra en estado Listo, el sistema operativo intercambia uno de los procesos bloqueados a disco, en la cola de Suspendidos. El sistema operativo trae otro proceso de la cola de Suspendidos o responde a una solicitud de un nuevo proceso. La ejecución continúa con los nuevos procesos que han llegado.

El objetivo del Swapping es la de mejorar el rendimiento de los procesos de vez en cuando.

**8) Plantee un esquema general de las tablas el SO, detalle la estructura de los procesos.**

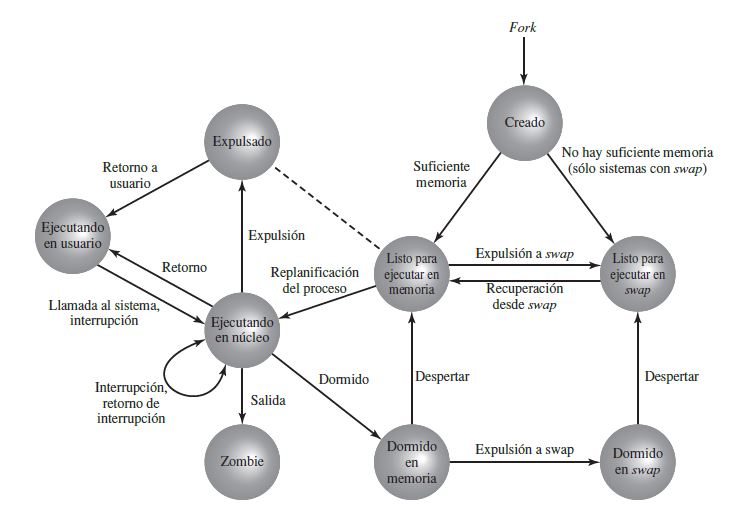


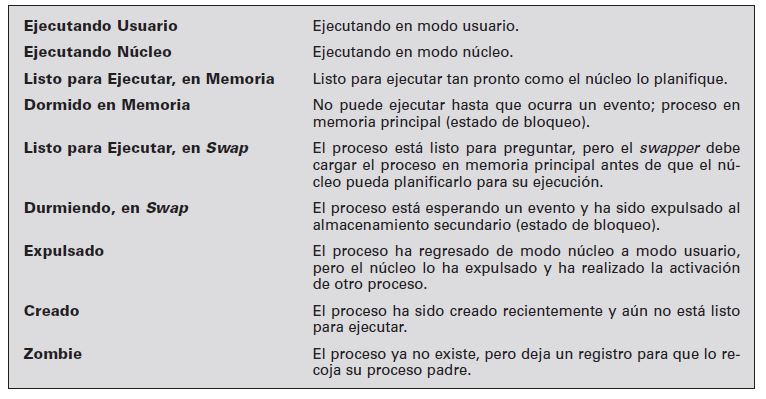
**9) Cuando se puede producir un intercambio de procesos. Explique cada opción.**

- Si no hay procesos listos, entonces al menos uno de los procesos bloqueados se transfiere al disco para hacer espacio para otro proceso que no se encuentra bloqueado.

- Cuando se necesita traer a MP un proceso de alta prioridad se puede llevar a MS un proceso de baja prioridad incluso si está en estado listo.

10) Dibuje un diagrama de estados de procesos en un SO UNIX de la familia SV4, explique cada estado.





**11) La creación de procesos en UNIX se realiza por medio de la llamada al sistema fork(). Cuando con un proceso solicita una llamada fork, el sistema operativo realiza las siguientes funciones [BACH86]:**

Para crear un nuevo proceso en el S.O. Unix se utiliza la llamada al sistema fork. Esta llamada hace que el proceso que la ejecuta se divida en dos procesos. Al proceso que ejecuta fork se le conoce como proceso padre y al nuevo proceso creado se le llama proceso hijo

1. Solicita la entrada en la tabla de procesos para el nuevo proceso.

2. Asigna un identificador de proceso único al proceso hijo.

3. Hace una copia de la imagen del proceso padre, con excepción de las regiones de memoria compartidas.

4. Incrementa el contador de cualquier fichero en posesión del padre, para reflejar el proceso adicional que ahora también posee dichos ficheros.

5. Asigna al proceso hijo el estado Listo para Ejecutar.

6. Devuelve el identificador del proceso hijo al proceso padre, y un valor 0 al proceso hijo.

#include <stdio.h>

#include <unistd.h>

#include <sys/types.h>

int main(void)

{

pid\_t proceso;

printf("proceso padre PID = %d comienza\n", getpid());

proceso = fork();

if (proceso > 0)

{

printf("Proceso creado, soy el proceso padre");

printf(" PID = %d \n", getpid());

}

else if (proceso == 0)

{

printf("Proceso creado, soy el proceso hijo");

printf(" PID = %d \n", getppid());

}

else

{

printf("\nError al crear el proceso hijo\n");

}

return 0;

}