



TP Procesos e Hilos – Sistemas Operativos

- FCyT, Concepción del Uruguay
- 3° Licenciatura En Sistemas de Información



Estudiantes: Cepeda,
Leandro

Costantini, Francisco

Drotto, Maximiliano

Profesor: Ing. Aguiar

Mg. Rapallini

1. Defina proceso.

Un proceso es un **programa en ejecución**, una **entidad** que consiste de un código de programa y un conjunto de datos asociados a dicho código, y suponiendo que el procesador comienza a ejecutar este código de programa, nos referimos a esta entidad en ejecución como un **proceso**.

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.

- 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) sí ha sido creado.
- Listo**: un proceso que se prepara para ejecutar cuando tenga oportunidad.
- Ejecutando**: el proceso está actualmente en ejecución. Para este capítulo asumimos que el computador tiene un único procesador, de forma que solo un proceso puede estar en este estado en un instante determinado.
- Bloqueado**: un proceso que no puede ejecutar hasta que se cumpla un evento determinado o se complete una operación E/S.
- 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.

- Se utiliza un área de la pila distinta a la pila del proceso debido a que se debe almacenar información sobre el proceso interrumpido para poder posteriormente reanudar el mismo con el contexto que tenía en el momento de la interrupción (contador de programa, PSW, registros).

5. ¿Cuáles son las actividades principales que el SO realiza en la gestión de procesos?

- Creación y terminación de procesos
- Planificación y activación de procesos
- Intercambio de procesos
- Sincronización de procesos y soporte para comunicación entre procesos
- Gestión de los bloques de control de proceso

6. Un proceso está en estado suspendido: ¿Cuáles son sus características y como pudo haber llegado a dicho estado?

Características de un proceso en estado **suspendido**:

- El proceso no está inmediatamente disponible para su ejecución.
- 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.
- 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.
- El proceso no puede ser recuperado de este estado hasta que el agente explícitamente así lo indique.

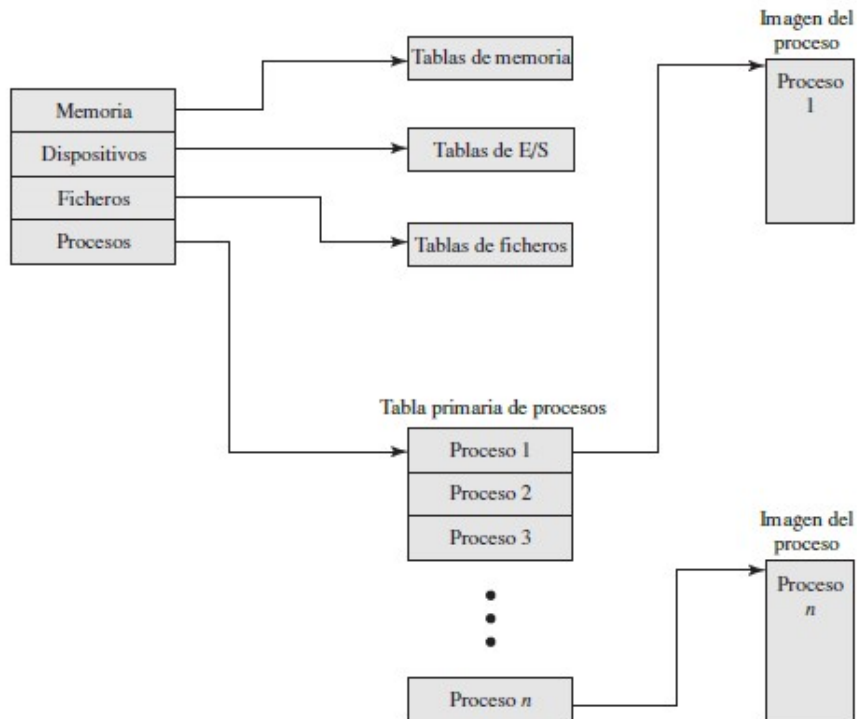
Un proceso puede pasar a estar suspendido porque:

<i>Swapping</i>	El sistema operativo necesita liberar suficiente memoria principal para traer un proceso en estado Listo de ejecución.
Otras razones del sistema operativo	El sistema operativo puede suspender un proceso en segundo plano o de utilidad o un proceso que se sospecha puede causar algún problema.
Solicitud interactiva del usuario	Un usuario puede desear suspender la ejecución de un programa con motivo de su depuración o porque está utilizando un recurso.
Temporización	Un proceso puede ejecutarse periódicamente (por ejemplo, un proceso monitor de estadísticas sobre el sistema) y puede suspenderse mientras espera el siguiente intervalo de ejecución.
Solicitud del proceso padre	Un proceso padre puede querer suspender la ejecución de un descendiente para examinar o modificar dicho proceso suspendido, o para coordinar la actividad de varios procesos descendientes.

7. Explique detalladamente que es el intercambio de procesos.

Un cambio de proceso ocurre cuando cierto proceso que está siendo procesado es reemplazado por otro, esto puede ocurrir en cualquier instante en el que el SO obtiene el control sobre el proceso actualmente en ejecución y hay tres causas para que esto ocurra: interrupción, trap y llamada al sistema.

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



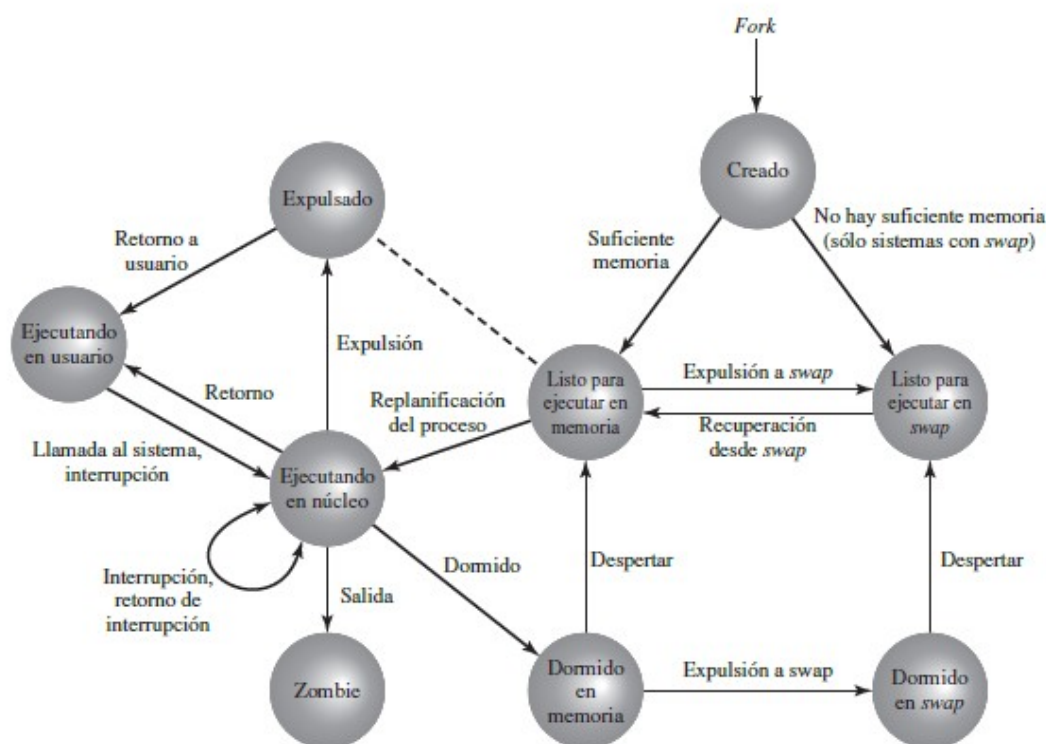
Estructura de los procesos:

Identificación del proceso
<p>Identificadores</p> <p>Identificadores numéricos que se pueden guardar dentro del bloque de control del proceso:</p> <ul style="list-style-type: none"> • identificadores del proceso • identificador del proceso que creó a este proceso (proceso padre) • identificador del usuario
Información de estado del procesador
<p>Registros visibles por el usuario</p> <p>Un registro visible por el usuario es aquel al que se puede hacer referencia por medio del lenguaje máquina que ejecuta el procesador cuando está en modo usuario. Normalmente, existen de 8 a 32 de estos registros, aunque determinadas implementaciones RISC tienen más de 100.</p> <p>Registros de estado y control</p> <p>Hay una gran variedad de registros del procesador que se utilizan para el control de operaciones. Estos incluyen:</p> <ul style="list-style-type: none"> • <i>Contador de programa</i>: contiene la dirección de la siguiente instrucción a ejecutar. • <i>Códigos de condición</i>: resultan de la operación lógica o aritmética más reciente (por ejemplo, signo, cero, acarreo, igual, desbordamiento). • <i>Información de estado</i>: incluyen los <i>flags</i> de interrupciones habilitadas/deshabilitadas, modo ejecución. <p>Puntero de pila</p> <p>Cada proceso tiene una o más pilas de sistema (LIFO) asociadas a él. Una pila se utiliza para almacenar los parámetros y las direcciones de retorno de los procedimientos y llamadas a sistema. Un puntero de pila apunta a la parte más alta de la pila.</p>
Información de control de proceso
<p>Información de estado y de planificación</p> <p>Esta es la información que el sistema operativo necesita para analizar las funciones de planificación. Elementos típicos de esta información son:</p> <ul style="list-style-type: none"> • <i>Estado del proceso</i>: indica si está listo o no el proceso para ser planificado para su ejecución (por ejemplo, Ejecutando, Listo, Esperando, Detenido). • <i>Prioridad</i>: uno o más campos que se pueden utilizar para escribir la prioridad de planificación del proceso. En algunos sistemas, se necesitan múltiples valores (por ejemplo, por-defecto, actual, mayor-disponible). • <i>Información relativa a planificación</i>: esto dependerá del algoritmo de planificación utilizado. Por ejemplo, la cantidad de tiempo que el proceso estaba esperando y la cantidad de tiempo que el proceso ha ejecutado la última vez que estuvo corriendo. • <i>Evento</i>: identificar el evento al cual el proceso está esperando para continuar su ejecución. <p>Estructuración de datos</p> <p>Un proceso puede estar enlazado con otro proceso en una cola, anillo o con cualquier otra estructura. Por ejemplo, todos los procesos en estado de espera por un nivel de prioridad en particular pueden estar enlazados en una cola. Un proceso puede mostrar una relación padre-hijo (creador-creado) con otro proceso. El bloque de control del proceso puede contener punteros a otros procesos para dar soporte a estas estructuras.</p> <p>Comunicación entre procesos</p> <p>Se pueden asociar diferentes <i>flags</i>, señales, y mensajes relativos a la comunicación entre dos procesos independientes. Alguna o toda esta información se puede mantener en el bloque de control de proceso (BCP).</p> <p>Privilegios de proceso</p> <p>Los procesos adquieren privilegios de acuerdo con la memoria a la que van a acceder y los tipos de instrucciones que se pueden ejecutar. Adicionalmente, los privilegios se pueden utilizar para usar utilidades de sistema o servicios.</p> <p>Gestión de memoria</p> <p>Esta sección puede incluir punteros a tablas de segmentos y/o páginas que describen la memoria virtual asignada a este proceso.</p> <p>Propia de recursos y utilización</p> <p>Se deben indicar los recursos controlados por el proceso, por ejemplo, ficheros abiertos. También se puede incluir un histórico de utilización del procesador o de otros recursos; esta información puede ser necesaria para el planificador.</p>

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

Mecanismo	Causa	Uso
Interrupción	Externa a la ejecución del proceso actualmente en ejecución.	Reacción ante un evento externo asíncrono.
Trap	Asociada a la ejecución de la instrucción actual.	Manejo de una condición de error o de excepción.
Llamada al sistema	Solicitud explícita.	Llamada a una función del sistema operativo.

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



Ejecutando usuario: ejecutando en modo usuario.

Ejecutando núcleo: ejecutando en modo núcleo.

Listo para ejecutar, en memoria: listo para ejecutar tan pronto como el núcleo lo planifique.

Dormido en memoria: no puede ejecutar hasta que ocurra un evento; proceso en memoria principal (estado de bloqueo).

Listo para ejecutar, en swap: el proceso está listo para preguntar, pero el swapper debe cargar el proceso en memoria principal antes de que el núcleo pueda planificarlo para su ejecución.

Durmiendo, en swap: el proceso está esperando un evento y ha sido expulsado al almacenamiento secundario (estado de bloqueo).

Expulsado: el proceso ha regresado de modo núcleo a modo usuario, pero el núcleo lo ha expulsado y ha realizado la activación de otro proceso.

Creado: el proceso ha sido creado recientemente y aún no está listo para ejecutar.

Zombie: el proceso ya no existe, pero deja un registro para que lo recoja su proceso padre.

11. En programación. ¿Cómo se crean los procesos en UNIX? De un ejemplo en lenguaje C.

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]:

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.

Todo este trabajo se realiza en **modo núcleo**, dentro del proceso padre. Cuando el núcleo ha completado estas funciones puede realizar cualquiera de las siguientes acciones, como parte de la rutina del activador:

- Continuar con el proceso padre:** el control vuelve a modo usuario en el punto en el que se realizó la llamada fork por parte del padre.
- Transferir el control al proceso hijo:** el proceso hijo comienza ejecutar en el mismo punto del código del padre, es decir en el punto de retorno de la llamada fork.
- Transferir el control a otro proceso:** ambos procesos, padre e hijo, permanecen en el estado Listos para Ejecutar.

Ejemplo de **fork** en C:

```
1  #include <sys/types.h>
2  #include <unistd.h>
3  #include <stdio.h>
4
5  int main(int argc, char *argv[])
6  {
7      pid_t pid;
8
9      if ( (pid=fork()) == 0 )
10     { /* hijo */
11         printf("Soy el hijo (%d, hijo de %d)\n", getpid(),
12             getppid());
13     }
14     else
15     { /* padre */
16         printf("Soy el padre (%d, hijo de %d)\n", getpid(),
17             getppid());
18     }
19
20     return 0;
21 }
```

En C, cuando el process ID (lo que devuelve el método getpid()) es igual a cero, es porque estamos en el proceso hijo, en cambio si el process ID es diferente a cero entonces estamos en el proceso padre.