

Licenciatura en Sistemas de Información

Trabajo Práctico N° 4

Procesos

Cátedra: Sistemas Operativos

Alumnos: Rondán Nicolás, Ruiz Mazzocato Nicolás

Universidad Autónoma de Entre Ríos

Facultad de Ciencia y Tecnología

Año: 2019

Procesos

1) Defina proceso.

El término proceso, en informática, se puede interpretar de diferentes maneras y, por lo tanto, otorgarle diferentes definiciones, entre las que se encuentran:

Un programa en ejecución.

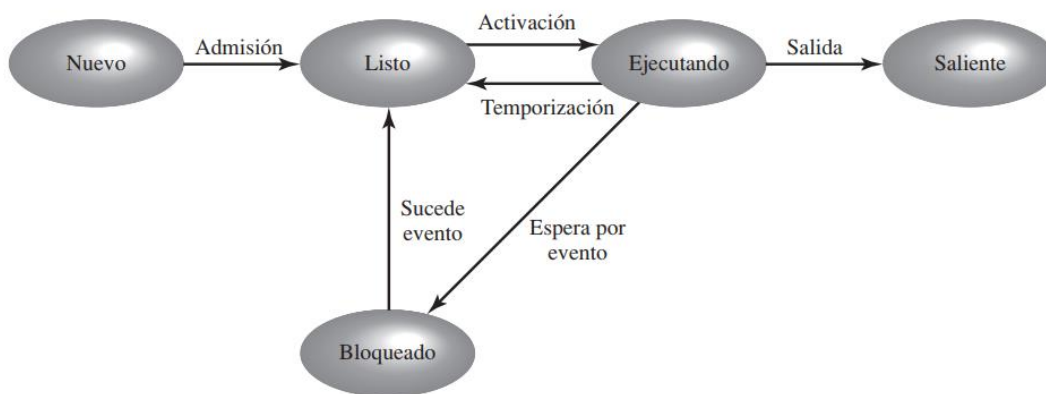
Una instancia de un programa ejecutado en un computador.

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.

También se puede pensar en un proceso como en 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.

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.

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) sí 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 utilizar un área de la pila distinta a la pila del proceso. Explique cuál es la razón.

Cuando ocurre una interrupción o llamada al sistema, el procesador se pone en modo núcleo y el control se pasa al sistema operativo. Para este fin, el contexto se salva (pila kernel) y se cambia de modo a una rutina del sistema operativo. Sin embargo, la ejecución continúa dentro del proceso de usuario actual. De esta forma, no se realiza un cambio de proceso, sino un cambio de modo dentro del mismo proceso. Si el sistema operativo, después de haber realizado su trabajo, determina que el proceso actual debe continuar con su ejecución, entonces el cambio de modo continúa con el programa interrumpido, dentro del mismo proceso.

5) ¿Cuáles son las actividades principales que el SO realizar 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.

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.

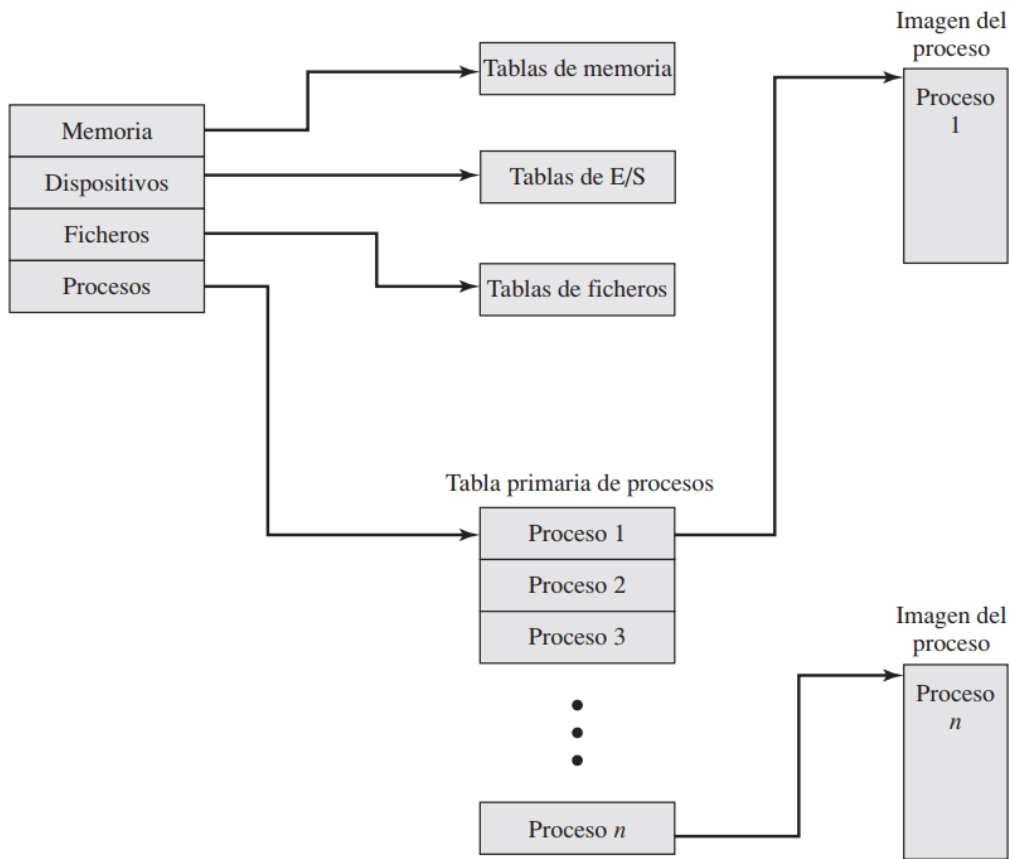
Razones por las cuales pudo haber llegado a ese estado:

- Swapping: 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.

El swapping (memoria de intercambio) implica mover 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. Esta es una lista de procesos existentes que han sido temporalmente expulsados de la memoria principal, o 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.

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



Las **tablas de memoria** se usan para mantener un registro tanto de la memoria principal (real) como de la secundaria (virtual). Parte de la memoria principal está reservada para el uso del sistema operativo; el resto está disponible para el uso de los procesos.

El sistema operativo debe utilizar las **tablas de E/S** para gestionar los dispositivos de E/S y los canales del computador.

Estructura de un proceso:

Como mínimo, un proceso debe incluir un programa o un conjunto de programas a ejecutar. Asociados con estos programas existen unas posiciones de memoria para los datos de variables locales y globales y de cualquier constante definida. Así, un proceso debe consistir en una cantidad suficiente de memoria para almacenar el programa y datos del mismo.

Adicionalmente, la ejecución típica de un programa incluye una pila que se utiliza para registrar las llamadas a procedimientos y los parámetros pasados entre dichos procedimientos. Por último, cada proceso está asociado a un número de atributos que son utilizados por el sistema operativo para controlar el proceso. Normalmente, el conjunto de estos atributos se denomina **bloque de control del proceso** (BCP). Nos podemos referir al conjunto de programa, datos, pila, y atributos, como la **imagen del proceso**.

- **Elementos típicos de la imagen de un proceso:**

Datos del usuario: la parte modificable del espacio de usuario. Puede incluir datos de programa, área de pila de usuario, y programas que puedan ser modificados.

Programa de usuario: el programa a ejecutar.

Pila de sistema: 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.

Bloque de control de proceso: datos necesarios para que el sistema operativo pueda controlar los procesos.

- **Elementos típicos de un bloque de control de proceso:**

Identificación del proceso

Identificadores: identificadores numéricos que se pueden guardar dentro del bloque de control del proceso:

- identificadores del proceso.
- identificador del proceso que creó a este proceso (proceso padre).
- identificador del usuario.

Información de estado del procesador

Registros visibles por el usuario: 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.

Registros de estado y control: Hay una gran variedad de registros del procesador que se utilizan para el control de operaciones. Estos incluyen:

- *Contador de programa:* contiene la dirección de la siguiente instrucción a ejecutar.
- *Códigos de condición:* resultan de la operación lógica o aritmética más reciente (por ejemplo, signo, cero, acarreo, igual, desbordamiento).
- *Información de estado:* incluyen los *flags* de interrupciones habilitadas/deshabilitadas, modo ejecución.

Puntero de pila: 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.

Información de control de proceso

Información de estado y de planificación: Esta es la información que el sistema operativo necesita para analizar las funciones de planificación.

Elementos típicos de esta información son:

- *Estado del proceso:* indica si está listo o no el proceso para ser planificado para su ejecución (por ejemplo, Ejecutando, Listo, Esperando, Detenido).
- *Prioridad:* 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).
- *Información relativa a planificación:* 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.
- *Evento:* identificar el evento al cual el proceso está esperando para continuar su ejecución.

Estructuración de datos

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.

Comunicación entre procesos

Se pueden asociar diferentes *flags*, 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).

Privilegios de proceso

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.

Gestión de memoria

Esta sección puede incluir punteros a tablas de segmentos y/o páginas que describen la memoria virtual asignada a este proceso.

Propia de recursos y utilización

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.

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

Un cambio de proceso puede ocurrir en cualquier instante en el que el sistema operativo obtiene el control sobre el proceso actualmente en ejecución. Dicho control se puede obtener mediante interrupciones, traps y llamadas al sistema. Dentro de las **interrupciones** podemos distinguir:

Interrupción de reloj: El sistema operativo determinar si el proceso en ejecución ha excedido o no la unidad máxima de tiempo de ejecución, denominada rodaja de tiempo (time slice). Esto es, una rodaja de tiempo es la máxima cantidad de tiempo que un proceso puede ejecutar antes de ser interrumpido. En dicho caso, este proceso se puede pasar al estado de Listo y se puede activar otro proceso.

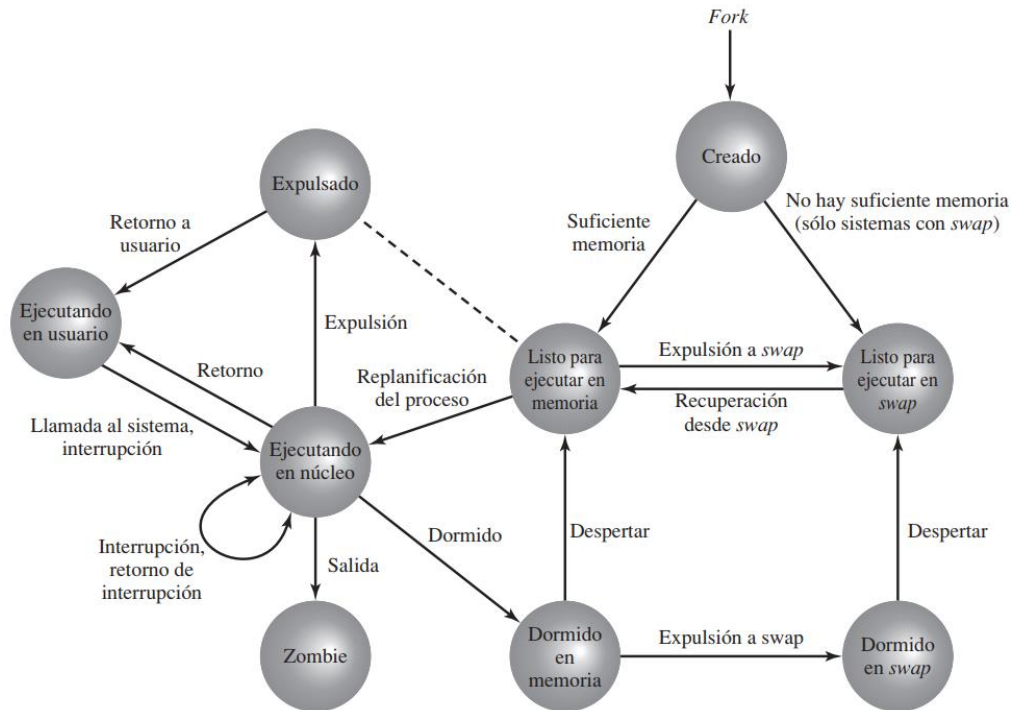
Interrupción de E/S: El sistema operativo determina qué acción de E/S ha ocurrido. Si la acción de E/S constituye un evento por el cual están esperando uno o más procesos, el sistema operativo mueve todos los procesos correspondientes al estado de Listos (y los procesos en estado Bloqueado/Suspendido al estado Listo/Suspendido). El sistema operativo puede decidir si reanuda la ejecución del proceso actualmente en estado Ejecutando o si lo expulsa para proceder con la ejecución de un proceso Listo de mayor prioridad.

Fallo de memoria: El procesador se encuentra con una referencia a una dirección de memoria virtual, a una palabra que no se encuentra en memoria principal. El sistema operativo debe traer el bloque (página o segmento) que contiene la referencia desde memoria secundaria a memoria principal. Después de que se solicita la operación de E/S para traer el bloque a memoria, el proceso que causó el fallo de memoria se pasa al estado de Bloqueado; el sistema operativo realiza un cambio de proceso y pone a ejecutar a otro proceso. Después de que el bloque de memoria solicitado se haya traído, el proceso pasará al estado Listo.

Con un **trap**, el sistema operativo conoce si una condición de error o de excepción es irreversible. Si es así, el proceso en ejecución se pone en el estado Saliente y se hace un cambio de proceso. De no ser así, el sistema operativo actuará dependiendo de la naturaleza del error y su propio diseño. Se puede intentar un procedimiento de recuperación o simplemente la notificación al usuario, pudiendo implicar tanto un cambio de proceso como la continuación de la ejecución del proceso actual.

Por último, el sistema operativo se puede activar por medio de una **llamada al sistema** procedente del programa en ejecución. Por ejemplo, si se está ejecutando un proceso y se ejecuta una operación que implica una llamada de E/S, como abrir un archivo. Esta llamada implica un salto a una rutina que es parte del código del sistema operativo. La realización de una llamada al sistema puede implicar en algunos casos que el proceso que la realiza pase al estado de Bloqueado.

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:

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:

1. 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.
2. 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.
3. Transferir el control a otro proceso. Ambos procesos, padre e hijo, permanecen en el estado Listos para Ejecutar.

Uso del 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 }
```