Trabajo Práctico N°3

**Procesos**

**1.-** En la figura se muestran los estados que puede tener un proceso en el sistema.

1. Enuncie las causas de las transiciones 1, 2, 3 y 4.

Bloqueado a Listo: El recurso que estaba esperando el proceso se completó por lo que ya está listo para ser ejecutado.

Listo a ejecutando: el sistema operativo le entrega el uso del procesador al proceso.

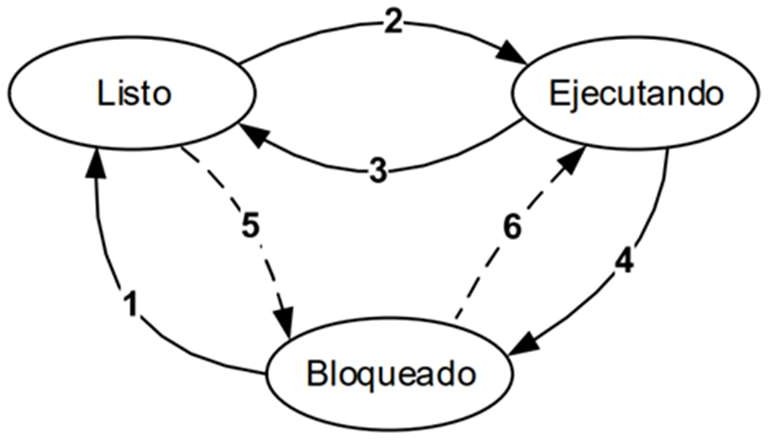
Ejecutando a listo: el sistema operativo decide quitarle el uso del procesador al proceso ejecutándose o voluntariamente el proceso decide abandonar voluntariamente el procesador.

Ejecutando a bloqueado: el proceso identifica que necesita un recurso y está a la espera del mismo.

1. ¿Podrían existir algunas circunstancias que provoquen las transiciones 5 y 6? Explique para cada caso, porqué sí o porqué no.

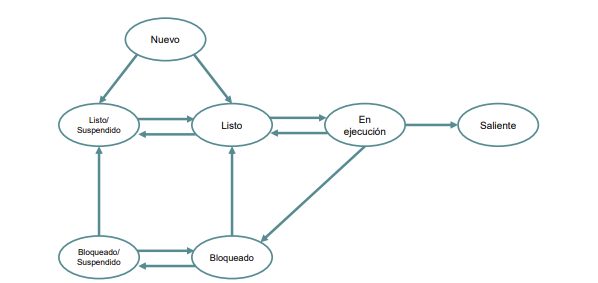
La transición 5 no puede llevarse a cabo ya que para que un proceso identifique que necesita un recurso tiene que estar en ejecución.

La transición 6 no puede llevarse a cabo ya que tiene que ser el SO el que decida llevarlo a ejecución y para eso previamente tiene que estar en la cola de listo.



**2.**- Algunos sistemas operativos, como UNIX System V, pueden llevar un proceso completo de memoria primaria a disco (swap out), y de disco a memoria primaria (swap in). También un usuario puede suspender su proceso en ejecución, el que es enviado directamente a disco. Estas acciones son llevadas a cabo por el proceso “swapper”.

1. Dibuje un diagrama de todos los estados por los que puede evolucionar un proceso en el sistema. Incluya los estados que puede tener cuando ocurren las acciones mencionadas y, además, los estados “nuevo” y “terminado”.



1. Enuncie las acciones o sucesos que provocan las transiciones entre estados.

Un proceso pasa de nuevo a listo en caso de que al ser creado exista espacio en la RAM o el SO decide asignarle un espacio en la misma, quitando otro proceso.

Un proceso pasa de nuevo a listo/suspendido cuando no hay espacio suficiente en la RAM y tiene que ser llevado a disco.

Un proceso pasa de listo a listo/suspendido cuando es necesario liberar RAM para otros procesos. La transición en el sentido contrario sucede cuando el SO decide pasar un proceso de disco a RAM.

Un proceso pasa de bloqueado a bloqueado/suspendido cuando es necesario liberar RAM para otros procesos porque, por ejemplo, puede llevarle varios ciclos de reloj obtener el recurso. En el sentido contrario, la transición porque se ha liberado memoria en RAM.

Un proceso pasa de bloqueado/suspendido a listo/suspendido porque ya obtuvo el recurso que estaba esperando.

Un proceso pasa de en ejecución a saliente cuando termina su ejecución.

**3.**- Describa los pasos que debe realizar el sistema operativo cuando un proceso termina por cualquiera de las razones estudiadas en la teoría.

1. Salvar el estado del procesador
2. Actualizar el bloque de datos del control del proceso actual.
3. Mover el proceso a la cola apropiada.
4. Seleccionar un nuevo proceso a ejecutar-
5. Actualizo el bloque de control del proceso elegido y actualizo a estado ejecutando.
6. Actualizar estructuras de datos de gestión de memoria.
7. Restaurar el estado del procesador que tenía el proceso seleccionado.

**4.**- Considerando el diagrama de estado de UNIX System V:

1. Muestre los estados por los que puede transitar un proceso en los siguientes escenarios: a1.Un proceso está ejecutando e invoca un “system call” que abre un archivo.

Ejecutando en usuario – Ejecutando en núcleo – Dormido en memoria.

a2.Un proceso está ejecutando y referencia una página que está en memoria secundaria. Luego el proceso es “swapped out”.

Ejecutando en usuario – Ejecutando en núcleo – Dormido en memoria-Dormido en swap

a3**.** Un proceso se encuentra dormido en memoria, pero recibe una señal de que debe continuar su ejecución.

Dormido en memoria – Listo en memoria – Ejecución en núcleo – Ejecución en usuario

a4. Un proceso se encuentra dormido en swap y es despertado.

Dormido en swap – Listo en swap

a5. Un proceso ejecutando en modo usuario recibe una interrupción de hardware (por ejemplo, del reloj).

Ejecutando en usuario – ejecutando en núcleo

a6. Un proceso es creado pero no hay suficiente memoria disponible en RAM.

Creado – Listo swap

1. En algunos textos se muestra el estado “Kernel running” con un lazo cerrado producido por una interrupción, como se muestra en la figura ¿En qué casos ocurre este retorno al mismo estado?

El estado "Kernel running" con un lazo cerrado ocurre en estos casos:

* Interrupciones anidadas: El kernel maneja una interrupción y otra de mayor prioridad lo interrumpe.
* Interrupciones en cascada: Una interrupción desencadena nuevas interrupciones antes de terminar.
* Excepciones durante interrupciones: Surge una excepción (como un fallo de página) mientras se atiende una interrupción.
* Alta carga de interrupciones: El kernel está constantemente procesando interrupciones sin retornar al modo usuario.

