

Sistemas Operativos 1

Procesos

Creación, estados e hilos

Edwin Salvador

29 de abril de 2015

Sesión 5

1 Creación y Terminación de procesos

2 Estados

- Modelos de estados de procesos
- Swapping

3 Procesos e hilos

- Funcionalidad de los Hilos
- Hilos de nivel de usuario y de nivel núcleo

4 Procesos en Windows y Ubuntu

Creación de procesos

- ¿Qué necesita el SO para poder crear un nuevo proceso?

Creación de procesos

- ¿Qué necesita el SO para poder crear un nuevo proceso? construir el BCP, reservar el espacio de direcciones de memoria principal para el proceso, asignar otros recursos.

Creación de procesos

- ¿Qué necesita el SO para poder crear un nuevo proceso? construir el BCP, reservar el espacio de direcciones de memoria principal para el proceso, asignar otros recursos.
- Existen cuatro eventos que llevan a la creación de un proceso:

Creación de procesos

- ¿Qué necesita el SO para poder crear un nuevo proceso? construir el BCP, reservar el espacio de direcciones de memoria principal para el proceso, asignar otros recursos.
- Existen cuatro eventos que llevan a la creación de un proceso:
 - Recepción y admisión de un nuevo trabajo.

Creación de procesos

- ¿Qué necesita el SO para poder crear un nuevo proceso? construir el BCP, reservar el espacio de direcciones de memoria principal para el proceso, asignar otros recursos.
- Existen cuatro eventos que llevan a la creación de un proceso:
 - Recepción y admisión de un nuevo trabajo.
 - La conexión del usuario provoca la creación de un proceso que ejecute el shell.

Creación de procesos

- ¿Qué necesita el SO para poder crear un nuevo proceso? construir el BCP, reservar el espacio de direcciones de memoria principal para el proceso, asignar otros recursos.
- Existen cuatro eventos que llevan a la creación de un proceso:
 - Recepción y admisión de un nuevo trabajo.
 - La conexión del usuario provoca la creación de un proceso que ejecute el shell.
 - Respuesta a una petición del usuario.

Creación de procesos

- ¿Qué necesita el SO para poder crear un nuevo proceso? construir el BCP, reservar el espacio de direcciones de memoria principal para el proceso, asignar otros recursos.
- Existen cuatro eventos que llevan a la creación de un proceso:
 - Recepción y admisión de un nuevo trabajo.
 - La conexión del usuario provoca la creación de un proceso que ejecute el shell.
 - Respuesta a una petición del usuario.
 - Un proceso solicita la creación de otro proceso.



Terminación de procesos

De igual manera existen varias razones que pueden llevar a la terminación de un proceso. ¿Cuáles pueden ser?

- Finalización normal.

Terminación de procesos

De igual manera existen varias razones que pueden llevar a la terminación de un proceso. ¿Cuáles pueden ser?

- Finalización normal.
- Límite de tiempo excedido (en ejecución).

Terminación de procesos

De igual manera existen varias razones que pueden llevar a la terminación de un proceso. ¿Cuáles pueden ser?

- Finalización normal.
- Límite de tiempo excedido (en ejecución).
- Límite de tiempo (en espera)

Terminación de procesos

De igual manera existen varias razones que pueden llevar a la terminación de un proceso. ¿Cuáles pueden ser?

- Finalización normal.
- Límite de tiempo excedido (en ejecución).
- Límite de tiempo (en espera)
- Memoria no disponible.

Terminación de procesos

De igual manera existen varias razones que pueden llevar a la terminación de un proceso. ¿Cuáles pueden ser?

- Finalización normal.
- Límite de tiempo excedido (en ejecución).
- Límite de tiempo (en espera)
- Memoria no disponible.
- Violaciones de frontera: acceder a direcciones de memoria no permitidas.

Terminación de procesos

De igual manera existen varias razones que pueden llevar a la terminación de un proceso. ¿Cuáles pueden ser?

- Finalización normal.
- Límite de tiempo excedido (en ejecución).
- Límite de tiempo (en espera)
- Memoria no disponible.
- Violaciones de frontera: acceder a direcciones de memoria no permitidas.
- Error de protección: permisos a recursos (archivos de solo lectura).

Terminación de procesos

De igual manera existen varias razones que pueden llevar a la terminación de un proceso. ¿Cuáles pueden ser?

- Finalización normal.
- Límite de tiempo excedido (en ejecución).
- Límite de tiempo (en espera)
- Memoria no disponible.
- Violaciones de frontera: acceder a direcciones de memoria no permitidas.
- Error de protección: permisos a recursos (archivos de solo lectura).
- Error aritmético: división por 0, overflow.

Terminación de procesos

De igual manera existen varias razones que pueden llevar a la terminación de un proceso. ¿Cuáles pueden ser?

- Finalización normal.
- Límite de tiempo excedido (en ejecución).
- Límite de tiempo (en espera)
- Memoria no disponible.
- Violaciones de frontera: acceder a direcciones de memoria no permitidas.
- Error de protección: permisos a recursos (archivos de solo lectura).
- Error aritmético: división por 0, overflow.
- Fallo de E/S: fichero no encontrado, permisos.

Terminación de procesos

De igual manera existen varias razones que pueden llevar a la terminación de un proceso. ¿Cuáles pueden ser?

- Finalización normal.
- Límite de tiempo excedido (en ejecución).
- Límite de tiempo (en espera)
- Memoria no disponible.
- Violaciones de frontera: acceder a direcciones de memoria no permitidas.
- Error de protección: permisos a recursos (archivos de solo lectura).
- Error aritmético: división por 0, overflow.
- Fallo de E/S: fichero no encontrado, permisos.
- Instrucción no válida.

Terminación de procesos

De igual manera existen varias razones que pueden llevar a la terminación de un proceso. ¿Cuáles pueden ser?

- Finalización normal.
- Límite de tiempo excedido (en ejecución).
- Límite de tiempo (en espera)
- Memoria no disponible.
- Violaciones de frontera: acceder a direcciones de memoria no permitidas.
- Error de protección: permisos a recursos (archivos de solo lectura).
- Error aritmético: división por 0, overflow.
- Fallo de E/S: fichero no encontrado, permisos.
- Instrucción no válida.
- Instrucción privilegiada: ejecución de instrucciones exclusivas al SO.

Terminación de procesos

- Uso inapropiado de datos.
- Intervención del operador por el SO.
- Terminación del proceso padre.
- Solicitud del proceso padre.

Contenido I

1 Creación y Terminación de procesos

2 Estados

- Modelos de estados de procesos
- Swapping

3 Procesos e hilos

- Funcionalidad de los Hilos
- Hilos de nivel de usuario y de nivel núcleo

4 Procesos en Windows y Ubuntu

Estados de un proceso

Los procesos evolucionan de manera cíclica entre diferentes estados. Estos estados pueden ser (los más comunes):

- **Nuevo** procesos que acaban de ser incluidos y el SO aún no los admite para ser ejecutados.

Estados de un proceso

Los procesos evolucionan de manera cíclica entre diferentes estados. Estos estados pueden ser (los más comunes):

- **Nuevo** procesos que acaban de ser incluidos y el SO aún no los admite para ser ejecutados.
- **Listo** procesos que cuentan con todos los recursos para comenzar o seguir su ejecución.

Estados de un proceso

Los procesos evolucionan de manera cíclica entre diferentes estados. Estos estados pueden ser (los más comunes):

- **Nuevo** procesos que acaban de ser incluidos y el SO aún no los admite para ser ejecutados.
- **Listo** procesos que cuentan con todos los recursos para comenzar o seguir su ejecución.
- **Ejecución** proceso que tiene el control del procesador. En sistemas con un solo procesador solo un proceso puede estar en este estado.

Estados de un proceso

Los procesos evolucionan de manera cíclica entre diferentes estados. Estos estados pueden ser (los más comunes):

- **Nuevo** procesos que acaban de ser incluidos y el SO aún no los admite para ser ejecutados.
- **Listo** procesos que cuentan con todos los recursos para comenzar o seguir su ejecución.
- **Ejecución** proceso que tiene el control del procesador. En sistemas con un solo procesador solo un proceso puede estar en este estado.
- **Bloqueado** procesos en espera de algún recurso o evento para continuar su ejecución.

Estados de un proceso

Los procesos evolucionan de manera cíclica entre diferentes estados. Estos estados pueden ser (los más comunes):

- **Nuevo** procesos que acaban de ser incluidos y el SO aún no los admite para ser ejecutados.
- **Listo** procesos que cuentan con todos los recursos para comenzar o seguir su ejecución.
- **Ejecución** proceso que tiene el control del procesador. En sistemas con un solo procesador solo un proceso puede estar en este estado.
- **Bloqueado** procesos en espera de algún recurso o evento para continuar su ejecución.
- **Terminado** procesos excluidos por el SO del grupo de procesos ejecutables. Un proceso pasa a terminado:

Estados de un proceso

Los procesos evolucionan de manera cíclica entre diferentes estados. Estos estados pueden ser (los más comunes):

- **Nuevo** procesos que acaban de ser incluidos y el SO aún no los admite para ser ejecutados.
- **Listo** procesos que cuentan con todos los recursos para comenzar o seguir su ejecución.
- **Ejecución** proceso que tiene el control del procesador. En sistemas con un solo procesador solo un proceso puede estar en este estado.
- **Bloqueado** procesos en espera de algún recurso o evento para continuar su ejecución.
- **Terminado** procesos excluidos por el SO del grupo de procesos ejecutables. Un proceso pasa a terminado:
 - terminación normal

Estados de un proceso

Los procesos evolucionan de manera cíclica entre diferentes estados. Estos estados pueden ser (los más comunes):

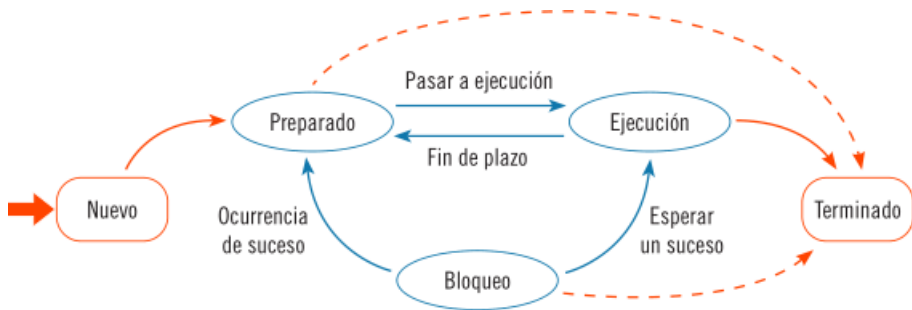
- **Nuevo** procesos que acaban de ser incluidos y el SO aún no los admite para ser ejecutados.
- **Listo** procesos que cuentan con todos los recursos para comenzar o seguir su ejecución.
- **Ejecución** proceso que tiene el control del procesador. En sistemas con un solo procesador solo un proceso puede estar en este estado.
- **Bloqueado** procesos en espera de algún recurso o evento para continuar su ejecución.
- **Terminado** procesos excluidos por el SO del grupo de procesos ejecutables. Un proceso pasa a terminado:
 - terminación normal
 - error irrecuperable

Estados de un proceso

Los procesos evolucionan de manera cíclica entre diferentes estados. Estos estados pueden ser (los más comunes):

- **Nuevo** procesos que acaban de ser incluidos y el SO aún no los admite para ser ejecutados.
- **Listo** procesos que cuentan con todos los recursos para comenzar o seguir su ejecución.
- **Ejecución** proceso que tiene el control del procesador. En sistemas con un solo procesador solo un proceso puede estar en este estado.
- **Bloqueado** procesos en espera de algún recurso o evento para continuar su ejecución.
- **Terminado** procesos excluidos por el SO del grupo de procesos ejecutables. Un proceso pasa a terminado:
 - terminación normal
 - error irrecuperable
 - terminación por otro proceso autorizado.

Estados de un proceso



Contenido I

1 Creación y Terminación de procesos

2 Estados

- Modelos de estados de procesos
- Swapping

3 Procesos e hilos

- Funcionalidad de los Hilos
- Hilos de nivel de usuario y de nivel núcleo

4 Procesos en Windows y Ubuntu

- La responsabilidad principal del SO es controlar la ejecución de los procesos, por lo tanto debe determinar el patrón de **entrelazado** para la ejecución y asignación de recursos a estos procesos.

- La responsabilidad principal del SO es controlar la ejecución de los procesos, por lo tanto debe determinar el patrón de **entrelazado** para la ejecución y asignación de recursos a estos procesos.
- El primer paso para el diseño de un SO es describir el comportamiento que se desea que tengan los procesos.

Modelos de estados de procesos

- La responsabilidad principal del SO es controlar la ejecución de los procesos, por lo tanto debe determinar el patrón de **entrelazado** para la ejecución y asignación de recursos a estos procesos.
- El primer paso para el diseño de un SO es describir el comportamiento que se desea que tengan los procesos.
- A continuación veremos dos modelos de estados de procesos: un modelo de dos estados y uno de cinco estados.

Modelo de procesos de dos estados

- Con este modelo podemos decir que un proceso estará en dos posibles estados **Ejecutando** y **No Ejecutando**.

Modelo de procesos de dos estados

- Con este modelo podemos decir que un proceso estará en dos posibles estados **Ejecutando** y **No Ejecutando**.
- Cuando el SO, crea un proceso, crea un BCP para este nuevo proceso e inserta el proceso en el sistema en estado *No Ejecutando*. Entonces, en este punto el proceso ya existe, es conocido por el SO pero está esperando su oportunidad de ser ejecutado.

Modelo de procesos de dos estados

- Con este modelo podemos decir que un proceso estará en dos posibles estados **Ejecutando** y **No Ejecutando**.
- Cuando el SO, crea un proceso, crea un BCP para este nuevo proceso e inserta el proceso en el sistema en estado *No Ejecutando*. Entonces, en este punto el proceso ya existe, es conocido por el SO pero está esperando su oportunidad de ser ejecutado.
- Cada cierto tiempo el proceso actualmente en ejecución se interrumpirá y pasará al estado *No Ejecutando*.

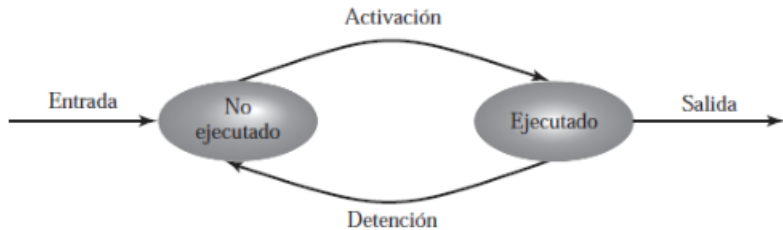
Modelo de procesos de dos estados

- Con este modelo podemos decir que un proceso estará en dos posibles estados **Ejecutando** y **No Ejecutando**.
- Cuando el SO, crea un proceso, crea un BCP para este nuevo proceso e inserta el proceso en el sistema en estado *No Ejecutando*. Entonces, en este punto el proceso ya existe, es conocido por el SO pero está esperando su oportunidad de ser ejecutado.
- Cada cierto tiempo el proceso actualmente en ejecución se interrumpirá y pasará al estado *No Ejecutando*.
- El **activador** seleccionará otro proceso para ejecutar y este pasará al estado *Ejecutando*.

Modelo de procesos de dos estados

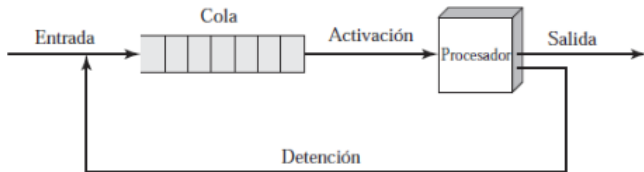
- Con este modelo podemos decir que un proceso estará en dos posibles estados **Ejecutando** y **No Ejecutando**.
- Cuando el SO, crea un proceso, crea un BCP para este nuevo proceso e inserta el proceso en el sistema en estado *No Ejecutando*. Entonces, en este punto el proceso ya existe, es conocido por el SO pero está esperando su oportunidad de ser ejecutado.
- Cada cierto tiempo el proceso actualmente en ejecución se interrumpirá y pasará al estado *No Ejecutando*.
- El **activador** seleccionará otro proceso para ejecutar y este pasará al estado *Ejecutando*.
- Los procesos en estado *No Ejecutando* deben ser almacenados en una especie de cola que contiene punteros a los BCP de cada proceso en este estado.

Diagrama de procesos con dos estados



(a) Diagrama de transiciones de estados

Modelo de colas con dos estados

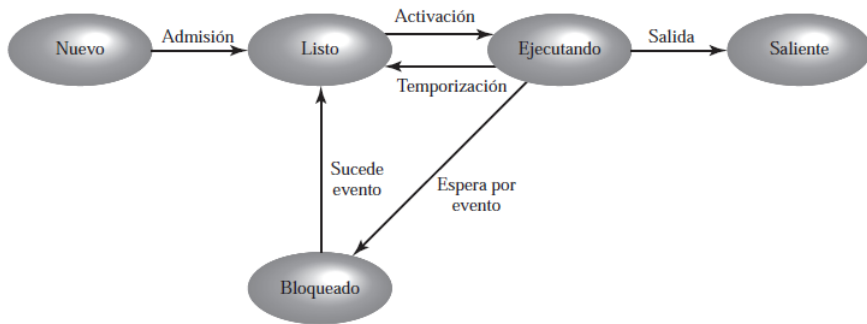


- ¿Qué pasará si el activador selecciona un proceso que no está listo para ser ejecutado? ¿Es eficiente utilizar únicamente dos estados?
- Esta cola es de tipo FIFO por lo tanto no toma en cuenta que procesos llevan más tiempo en la cola o que procesos están bloqueados esperando alguna operación de E/S. Por esta razón este modelo de dos estados no es eficiente.

Modelo de procesos de cinco estados

- Un modelo más eficiente podría dividir el estado *No Ejecutando* en dos: *Listo* y *Bloqueado*.
- Este modelo añade también un estado *Nuevo* y otro *Saliente*.
- Entonces los cinco estados son:
 - Ejecutando
 - Listo
 - Bloqueado (en espera de evento u operación E/S)
 - Nuevo (BCP creado pero aún no cargado en memoria principal).
 - Saliente (proceso detenido o abortado)
- ¿Qué ventajas nos presentará este modelo?
- ¿Cuáles son las posibles transiciones entre los estados?

Diagrama de modelo de procesos de cinco estados



Posibles transiciones entre estados

Modelo de procesos de cinco estados

Las posibles transiciones entre los 5 estados son:

- Null \rightarrow Nuevo

Posibles transiciones entre estados

Modelo de procesos de cinco estados

Las posibles transiciones entre los 5 estados son:

- Null \rightarrow Nuevo
- Nuevo \rightarrow Listo

Posibles transiciones entre estados

Modelo de procesos de cinco estados

Las posibles transiciones entre los 5 estados son:

- Null \rightarrow Nuevo
- Nuevo \rightarrow Listo
- Listo \rightarrow Ejecutando

Posibles transiciones entre estados

Modelo de procesos de cinco estados

Las posibles transiciones entre los 5 estados son:

- Null \rightarrow Nuevo
- Nuevo \rightarrow Listo
- Listo \rightarrow Ejecutando
- Ejecutando \rightarrow Saliente

Posibles transiciones entre estados

Modelo de procesos de cinco estados

Las posibles transiciones entre los 5 estados son:

- Null \rightarrow Nuevo
- Nuevo \rightarrow Listo
- Listo \rightarrow Ejecutando
- Ejecutando \rightarrow Saliente
- Ejecutando \rightarrow Listo

Posibles transiciones entre estados

Modelo de procesos de cinco estados

Las posibles transiciones entre los 5 estados son:

- Null \rightarrow Nuevo
- Nuevo \rightarrow Listo
- Listo \rightarrow Ejecutando
- Ejecutando \rightarrow Saliente
- Ejecutando \rightarrow Listo
- Ejecutando \rightarrow Bloqueado

Posibles transiciones entre estados

Modelo de procesos de cinco estados

Las posibles transiciones entre los 5 estados son:

- Null \rightarrow Nuevo
- Nuevo \rightarrow Listo
- Listo \rightarrow Ejecutando
- Ejecutando \rightarrow Saliente
- Ejecutando \rightarrow Listo
- Ejecutando \rightarrow Bloqueado
- Listo \rightarrow Saliente

Posibles transiciones entre estados

Modelo de procesos de cinco estados

Las posibles transiciones entre los 5 estados son:

- Null \rightarrow Nuevo
- Nuevo \rightarrow Listo
- Listo \rightarrow Ejecutando
- Ejecutando \rightarrow Saliente
- Ejecutando \rightarrow Listo
- Ejecutando \rightarrow Bloqueado
- Listo \rightarrow Saliente
- Bloqueado \rightarrow Saliente

Ejecución de procesos con modelo de cinco estados

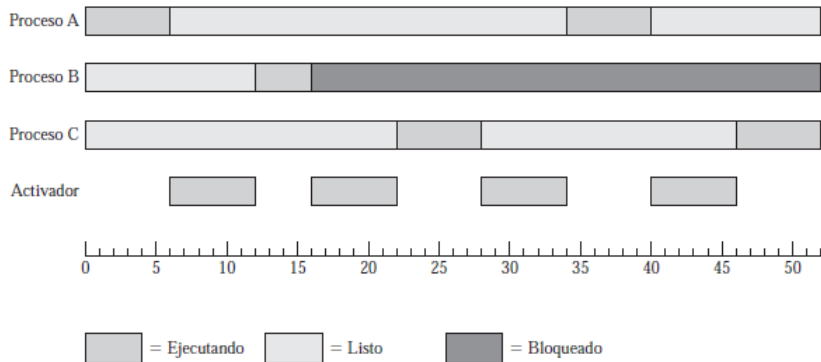
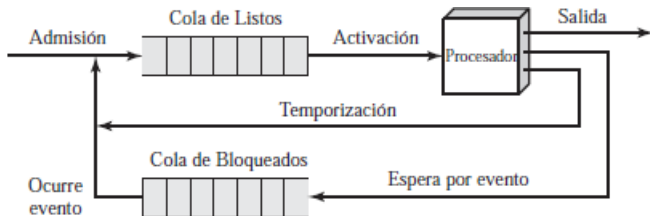


Figura: Estados de los procesos A, B y C

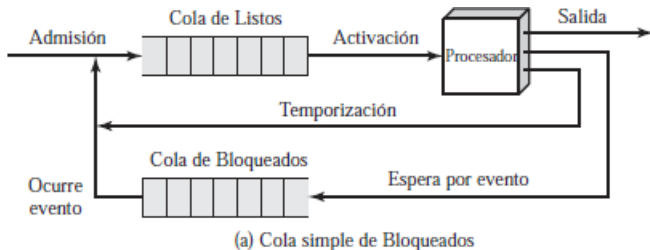
Modelo colas con cinco estados



(a) Cola simple de Bloqueados

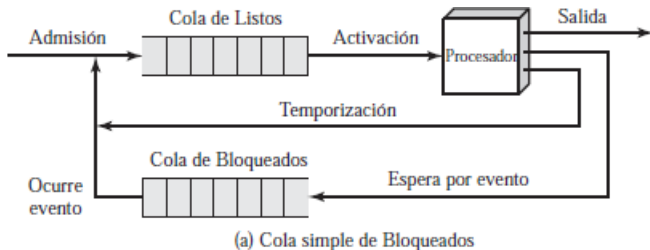
- En este modelo los estados *Listo* y *Bloqueado* son manejados por colas independientes.

Modelo colas con cinco estados



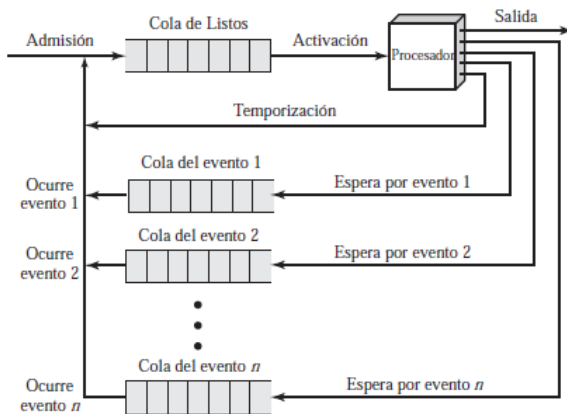
- En este modelo los estados *Listo* y *Bloqueado* son manejados por colas independientes.
- Cuando el SO debe seleccionar el siguiente proceso a ejecutar busca en la cola de *Listos* (**eficiente**). Podría existir un esquema de prioridades (una cola por cada prioridad) o simplemente un esquema FIFO.

Modelo colas con cinco estados



- En este modelo los estados *Listo* y *Bloqueado* son manejados por colas independientes.
- Cuando el SO debe seleccionar el siguiente proceso a ejecutar busca en la cola de *Listos* (**eficiente**). Podría existir un esquema de prioridades (una cola por cada prioridad) o simplemente un esquema FIFO.
- Cuando sucede algún evento el SO debe recorrer la cola de *Bloqueados* en busca de los procesos que estén esperando por este evento (**ineficiente**).

Modelo de múltiples colas de bloqueados



(b) Múltiples colas de Bloqueados

En algunos SO la cola de procesos *Bloqueados* podría contener miles de procesos por lo que sería más eficiente manejar **una cola por cada evento**.

Contenido I

1 Creación y Terminación de procesos

2 Estados

- Modelos de estados de procesos
- Swapping

3 Procesos e hilos

- Funcionalidad de los Hilos
- Hilos de nivel de usuario y de nivel núcleo

4 Procesos en Windows y Ubuntu

Swapping (procesos suspendidos)

- A pesar de que el modelo de múltiples colas de *Bloqueados* es más eficiente, la diferencia entre la velocidad del procesador y los dispositivos E/S podría ocasionar que todos los procesos lleguen a estar en un estado *Bloqueado*.

Swapping (procesos suspendidos)

- A pesar de que el modelo de múltiples colas de *Bloqueados* es más eficiente, la diferencia entre la velocidad del procesador y los dispositivos E/S podría ocasionar que todos los procesos lleguen a estar en un estado *Bloqueado*.
- ¿Qué pasaría si tenemos miles de procesos en estado *Bloqueado*?

Swapping (procesos suspendidos)

- A pesar de que el modelo de múltiples colas de *Bloqueados* es más eficiente, la diferencia entre la velocidad del procesador y los dispositivos E/S podría ocasionar que todos los procesos lleguen a estar en un estado *Bloqueado*.
- ¿Qué pasaría si tenemos miles de procesos en estado *Bloqueado*? podrían ocupar todo el espacio en memoria principal y el procesador no sería capaz de aceptar nuevos procesos (estaría ocioso).

Swapping (procesos suspendidos)

- A pesar de que el modelo de múltiples colas de *Bloqueados* es más eficiente, la diferencia entre la velocidad del procesador y los dispositivos E/S podría ocasionar que todos los procesos lleguen a estar en un estado *Bloqueado*.
- ¿Qué pasaría si tenemos miles de procesos en estado *Bloqueado*? podrían ocupar todo el espacio en memoria principal y el procesador no sería capaz de aceptar nuevos procesos (estaría ocioso).
- Podríamos aumentar la memoria pero esto aumentaría el costo del sistema. Y es importante notar que debido a las grandes memorias disponibles, la tendencia es a ejecutar **procesos más grandes y no más procesos**.

Swapping (procesos suspendidos)

- Una solución para lograr aceptar más procesos y disminuir el tiempo ocioso del procesador es utilizar una técnica llamada *swapping* (memoria de intercambio).

Swapping (procesos suspendidos)

- Una solución para lograr aceptar más procesos y disminuir el tiempo ocioso del procesador es utilizar una técnica llamada *swapping* (memoria de intercambio).
- El swapping consiste en intercambiar parte o todo el proceso de memoria principal a disco.

Swapping (procesos suspendidos)

- Una solución para lograr aceptar más procesos y disminuir el tiempo ocioso del procesador es utilizar una técnica llamada *swapping* (memoria de intercambio).
- El swapping consiste en intercambiar parte o todo el proceso de memoria principal a disco.
- El swapping añade el estado *Suspendido*

Swapping (procesos suspendidos)

- Una solución para lograr aceptar más procesos y disminuir el tiempo ocioso del procesador es utilizar una técnica llamada *swapping* (memoria de intercambio).
- El swapping consiste en intercambiar parte o todo el proceso de memoria principal a disco.
- El swapping añade el estado *Suspendido*
- Si no existen procesos en memoria principal en estado *Listo*, el SO mueve uno de estos procesos *Bloqueados* a disco (a la cola de procesos *Suspendidos*).

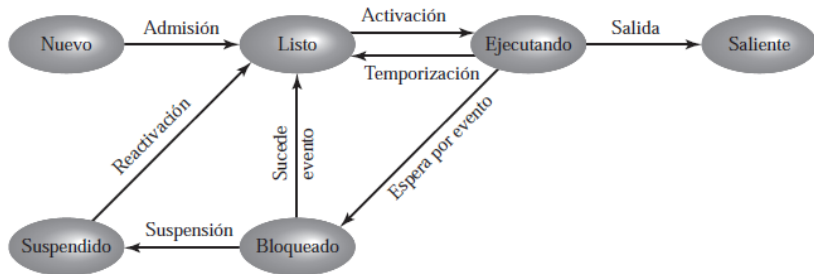
Swapping (procesos suspendidos)

- Una solución para lograr aceptar más procesos y disminuir el tiempo ocioso del procesador es utilizar una técnica llamada *swapping* (memoria de intercambio).
- El swapping consiste en intercambiar parte o todo el proceso de memoria principal a disco.
- El swapping añade el estado *Suspendido*
- Si no existen procesos en memoria principal en estado *Listo*, el SO mueve uno de estos procesos *Bloqueados* a disco (a la cola de procesos *Suspendidos*).
- Así el SO puede continuar con la creación de nuevos procesos o traer procesos de la cola de *Suspendidos* que puedan continuar siendo ejecutados.

Swapping (procesos suspendidos)

- Una solución para lograr aceptar más procesos y disminuir el tiempo ocioso del procesador es utilizar una técnica llamada *swapping* (memoria de intercambio).
- El swapping consiste en intercambiar parte o todo el proceso de memoria principal a disco.
- El swapping añade el estado *Suspendido*
- Si no existen procesos en memoria principal en estado *Listo*, el SO mueve uno de estos procesos *Bloqueados* a disco (a la cola de procesos *Suspendidos*).
- Así el SO puede continuar con la creación de nuevos procesos o traer procesos de la cola de *Suspendidos* que puedan continuar siendo ejecutados.
- Esta técnica generalmente mejora el rendimiento del sistema.

Swapping (procesos suspendidos)



Modelo de estados con swapping

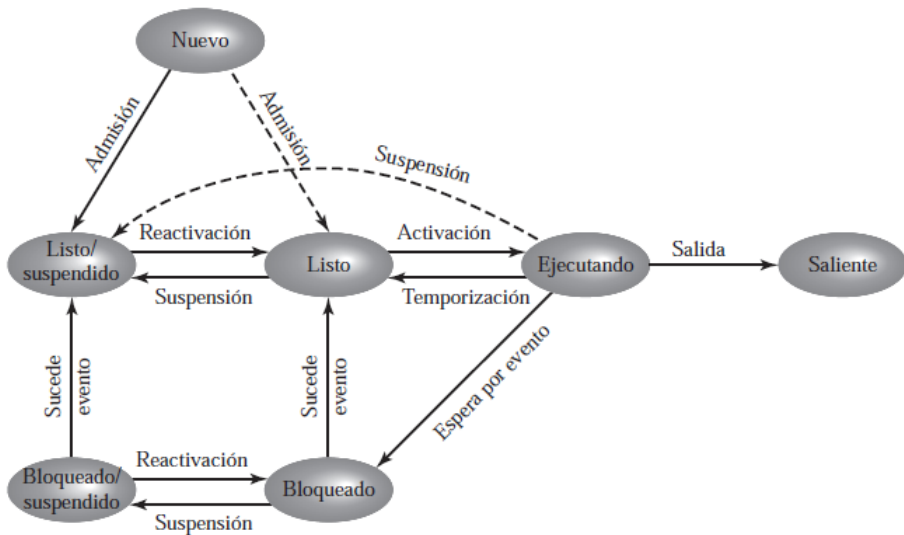
Para que el SO pueda determinar más fácilmente que proceso debe traer de la cola de *Suspendidos* se necesitan los siguientes estados:

- Listo
- Bloqueado
- Bloqueado/Suspendido
- Listo/Suspendido

¿Cuáles son las posibles transiciones entre estados en este modelo?

Modelo de estados con swapping

Listo/Suspendido y Bloqueado/Suspendido



Transiciones con swapping (dos estados suspendidos)

La figura anterior nos muestra las posibles transiciones entre estados cuando se utilizan dos estados suspendidos. Estas transiciones son:

- **Bloqueado** → **Bloqueado/Suspendido**
- **Bloqueado/Suspendido** → **Listo/Suspendido**
- **Listo/Suspendido** → **Listo**
- **Listo** → **Listo/Suspendido** (liberar memoria o prioridad baja)
- **Nuevo** → **Listo/Suspendido** o **Listo** (depende espacio en memoria)
- **Bloqueado/Suspendido** → **Bloqueado** (prioridad alta o evento pronto a ocurrir)
- **Ejecutando** → **Listo/Suspendido** (liberar memoria)
- **Cualquier estado** → **Saliente**

1 Creación y Terminación de procesos

2 Estados

- Modelos de estados de procesos
- Swapping

3 Procesos e hilos

- Funcionalidad de los Hilos
- Hilos de nivel de usuario y de nivel núcleo

4 Procesos en Windows y Ubuntu

- Hasta ahora conocemos a un proceso como poseedor de tres características:

- Hasta ahora conocemos a un proceso como poseedor de tres características:
 - Es un conjunto de instrucciones.

- Hasta ahora conocemos a un proceso como poseedor de tres características:
 - Es un conjunto de instrucciones.
 - Puede obtener el control de recursos (memoria principal, canales y dispositivos E/S y archivos)

- Hasta ahora conocemos a un proceso como poseedor de tres características:
 - Es un conjunto de instrucciones.
 - Puede obtener el control de recursos (memoria principal, canales y dispositivos E/S y archivos)
 - Puede estar en diferentes estados de ejecución (ejecutando, listo, etc) puede ser planificada y activada por el SO.

- Hasta ahora conocemos a un proceso como poseedor de tres características:
 - Es un conjunto de instrucciones.
 - Puede obtener el control de recursos (memoria principal, canales y dispositivos E/S y archivos)
 - Puede estar en diferentes estados de ejecución (ejecutando, listo, etc) puede ser planificada y activada por el SO.
- Estas características son la esencia de un proceso.

- Hasta ahora conocemos a un proceso como poseedor de tres características:
 - Es un conjunto de instrucciones.
 - Puede obtener el control de recursos (memoria principal, canales y dispositivos E/S y archivos)
 - Puede estar en diferentes estados de ejecución (ejecutando, listo, etc) puede ser planificada y activada por el SO.
- Estas características son la esencia de un proceso.
- Para distinguir estas hacer una distinción definimos a la unidad que se **activa** como **hilo (thread)**, mientras que la unidad de propiedad de **recursos** la definimos como **proceso** o **tarea**.

Multihilo

- Multihilo es la capacidad de un SO de dar soporte a múltiples hilos de ejecución en un solo proceso.

Multihilo

- Multihilo es la capacidad de un SO de dar soporte a múltiples hilos de ejecución en un solo proceso.
- No todos los SO soportan multihilo. Ejemplo:

Multihilo

- Multihilo es la capacidad de un SO de dar soporte a múltiples hilos de ejecución en un solo proceso.
- No todos los SO soportan multihilo. Ejemplo:
 - MS-DOS = 1 proceso de usuario y 1 único hilo.

Multihilo

- Multihilo es la capacidad de un SO de dar soporte a múltiples hilos de ejecución en un solo proceso.
- No todos los SO soportan multihilo. Ejemplo:
 - MS-DOS = 1 proceso de usuario y 1 único hilo.
 - Otros SO y algunas versiones de UNIX = múltiples procesos de usuario, pero 1 solo hilo.

- Multihilo es la capacidad de un SO de dar soporte a múltiples hilos de ejecución en un solo proceso.
- No todos los SO soportan multihilo. Ejemplo:
 - MS-DOS = 1 proceso de usuario y 1 único hilo.
 - Otros SO y algunas versiones de UNIX = múltiples procesos de usuario, pero 1 solo hilo.
- ¿Cuál es la diferencia entre varios procesos de un solo hilo y un proceso de varios hilos?

- Multihilo es la capacidad de un SO de dar soporte a múltiples hilos de ejecución en un solo proceso.
- No todos los SO soportan multihilo. Ejemplo:
 - MS-DOS = 1 proceso de usuario y 1 único hilo.
 - Otros SO y algunas versiones de UNIX = múltiples procesos de usuario, pero 1 solo hilo.
- ¿Cuál es la diferencia entre varios procesos de un solo hilo y un proceso de varios hilos?
- Windows, Solaris, Mac OS X, Linux entre otros soportan múltiples procesos y cada uno con múltiples hilos.

- En un entorno multihilo, un proceso es una unidad de asignación de recursos y una unidad de protección.
- Dentro de un proceso puede haber uno o más hilos, cada uno con:
 - Un estado de ejecución,

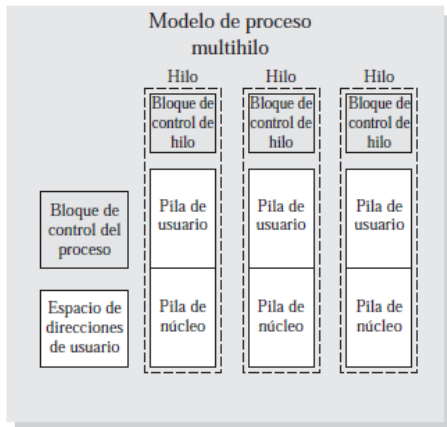
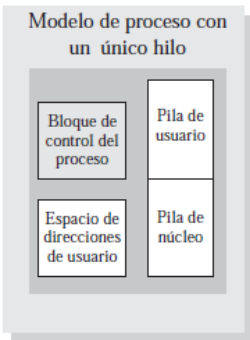
- En un entorno multihilo, un proceso es una unidad de asignación de recursos y una unidad de protección.
- Dentro de un proceso puede haber uno o más hilos, cada uno con:
 - Un estado de ejecución,
 - Un contexto de hilo (que se almacena cuando el hilo no está en ejecución).

- En un entorno multihilo, un proceso es una unidad de asignación de recursos y una unidad de protección.
- Dentro de un proceso puede haber uno o más hilos, cada uno con:
 - Un estado de ejecución,
 - Un contexto de hilo (que se almacena cuando el hilo no está en ejecución).
 - Una pila de ejecución.

- En un entorno multihilo, un proceso es una unidad de asignación de recursos y una unidad de protección.
- Dentro de un proceso puede haber uno o más hilos, cada uno con:
 - Un estado de ejecución,
 - Un contexto de hilo (que se almacena cuando el hilo no está en ejecución).
 - Una pila de ejecución.
 - Espacio de almacenamiento para variables locales.

- En un entorno multihilo, un proceso es una unidad de asignación de recursos y una unidad de protección.
- Dentro de un proceso puede haber uno o más hilos, cada uno con:
 - Un estado de ejecución,
 - Un contexto de hilo (que se almacena cuando el hilo no está en ejecución).
 - Una pila de ejecución.
 - Espacio de almacenamiento para variables locales.
 - Acceso a la memoria y recursos de su proceso, compartido con todos los hilos de su mismo proceso.

Diferencia entre proceso e hilo



- Multihilo: existe un solo BCP y un espacio de direcciones de usuario para el proceso, pero hay pilas separadas para cada hilo, y un BCP para cada hilo (registros, prioridad, etc).

- Todos los hilos de un proceso:
 - comparten el estado y los recursos de ese proceso.

- Todos los hilos de un proceso:
 - comparten el estado y los recursos de ese proceso.
 - residen en el mismo espacio de direcciones

- Todos los hilos de un proceso:
 - comparten el estado y los recursos de ese proceso.
 - residen en el mismo espacio de direcciones
 - tienen acceso a los mismos datos.

Beneficios de los hilos

Los mayores beneficios provienen de las consecuencias del rendimiento:

- Es más rápido crear un nuevo hilo en un proceso que crear un nuevo proceso (10-100 veces más rápido).

Beneficios de los hilos

Los mayores beneficios provienen de las consecuencias del rendimiento:

- Es más rápido crear un nuevo hilo en un proceso que crear un nuevo proceso (10-100 veces más rápido).
- Un hilo se finaliza más rápido que un proceso.

Beneficios de los hilos

Los mayores beneficios provienen de las consecuencias del rendimiento:

- Es más rápido crear un nuevo hilo en un proceso que crear un nuevo proceso (10-100 veces más rápido).
- Un hilo se finaliza más rápido que un proceso.
- Es más rápido cambiar entre dos hilos del **mismo** proceso.

Beneficios de los hilos

Los mayores beneficios provienen de las consecuencias del rendimiento:

- Es más rápido crear un nuevo hilo en un proceso que crear un nuevo proceso (10-100 veces más rápido).
- Un hilo se finaliza más rápido que un proceso.
- Es más rápido cambiar entre dos hilos del **mismo** proceso.
- Mejoran la eficiencia de la comunicación entre diferentes programas.

- Entonces, si se desea crear una aplicación como un conjunto de unidades de ejecución relacionadas, es mucho más eficiente hacerlo con un conjunto de hilos que con un conjunto de procesos independientes.

- Entonces, si se desea crear una aplicación como un conjunto de unidades de ejecución relacionadas, es mucho más eficiente hacerlo con un conjunto de hilos que con un conjunto de procesos independientes.
- Un ejemplo de aplicación que puede hacer uso de hilos es un servidor de archivos. Cada vez que se realiza una petición de archivos, el gestor de archivos puede ejecutar un nuevo hilo. Así en un ambiente **multiprocesador** se pueden ejecutar simultáneamente múltiples hilos del mismo proceso en diferentes procesadores. Y los hilos pueden compartir archivos de datos y coordinar acciones de manera más rápida.

- Entonces, si se desea crear una aplicación como un conjunto de unidades de ejecución relacionadas, es mucho más eficiente hacerlo con un conjunto de hilos que con un conjunto de procesos independientes.
- Un ejemplo de aplicación que puede hacer uso de hilos es un servidor de archivos. Cada vez que se realiza una petición de archivos, el gestor de archivos puede ejecutar un nuevo hilo. Así en un ambiente **multiprocesador** se pueden ejecutar simultáneamente múltiples hilos del mismo proceso en diferentes procesadores. Y los hilos pueden compartir archivos de datos y coordinar acciones de manera más rápida.
- ¿Ejemplos? Hacer grupos y escribir la mayor cantidad de ejemplos sobre programas que utilizan hilos durante su ejecución.

¿Cómo se utilizan los hilos en un sistema multiprocesador?

- **Trabajo en primer plano y en segundo plano:** Un programa de hojas de cálculo (Excel) puede utilizar un hilo para mostrar menús mientras otro hilo ejecuta los mandatos de usuario y actualiza la hoja de cálculo. Así se puede empezar un nuevo mandato antes de terminar el anterior y se percibe una mayor velocidad de la aplicación.

¿Cómo se utilizan los hilos en un sistema multiprocesador?

- **Trabajo en primer plano y en segundo plano:** Un programa de hojas de cálculo (Excel) puede utilizar un hilo para mostrar menús mientras otro hilo ejecuta los mandatos de usuario y actualiza la hoja de cálculo. Así se puede empezar un nuevo mandato antes de terminar el anterior y se percibe una mayor velocidad de la aplicación.
- **Procesamiento asíncrono:** Un procesador de texto (Word) utiliza **tareas asíncronas** que pueden implementarse como hilos. Se puede crear un hilo cuyo único trabajo sea crear una copia de seguridad cada minuto y guardarla en disco.

¿Cómo se utilizan los hilos en un sistema multiprocesador?

- **Trabajo en primer plano y en segundo plano:** Un programa de hojas de cálculo (Excel) puede utilizar un hilo para mostrar menús mientras otro hilo ejecuta los mandatos de usuario y actualiza la hoja de cálculo. Así se puede empezar un nuevo mandato antes de terminar el anterior y se percibe una mayor velocidad de la aplicación.
- **Procesamiento asíncrono:** Un procesador de texto (Word) utiliza **tareas asíncronas** que pueden implementarse como hilos. Se puede crear un hilo cuyo único trabajo sea crear una copia de seguridad cada minuto y guardarla en disco.
- **Velocidad de ejecución:** un proceso multihilo puede computar una serie de datos mientras lee los siguientes datos de un dispositivo. Así mientras un hilo está bloqueado por una operación de E/S, otro hilo puede estar ejecutando.

¿Cómo se utilizan los hilos en un sistema multiprocesador?

- **Trabajo en primer plano y en segundo plano:** Un programa de hojas de cálculo (Excel) puede utilizar un hilo para mostrar menús mientras otro hilo ejecuta los mandatos de usuario y actualiza la hoja de cálculo. Así se puede empezar un nuevo mandato antes de terminar el anterior y se percibe una mayor velocidad de la aplicación.
- **Procesamiento asíncrono:** Un procesador de texto (Word) utiliza **tareas asíncronas** que pueden implementarse como hilos. Se puede crear un hilo cuyo único trabajo sea crear una copia de seguridad cada minuto y guardarla en disco.
- **Velocidad de ejecución:** un proceso multihilo puede computar una serie de datos mientras lee los siguientes datos de un dispositivo. Así mientras un hilo está bloqueado por una operación de E/S, otro hilo puede estar ejecutando.
- **Estructura modular de programas:** Los programas con varias tareas o con varias fuentes y destinos de E/S se pueden implementar más fácilmente usando hilos.

- Debido a que todos los hilos de un proceso comparten el mismo espacio de direcciones que su proceso, al suspender el proceso, todos los hilos se suspenden al mismo tiempo. De igual manera cuando se finaliza un proceso, se finalizan todos los hilos de ese proceso.

Contenido I

1 Creación y Terminación de procesos

2 Estados

- Modelos de estados de procesos
- Swapping

3 Procesos e hilos

- Funcionalidad de los Hilos
- Hilos de nivel de usuario y de nivel núcleo

4 Procesos en Windows y Ubuntu

Funcionalidad de los Hilos

Estados de los hilos

- Los principales estados de los hilos son Ejecutando, Listo y Bloqueado.

Funcionalidad de los Hilos

Estados de los hilos

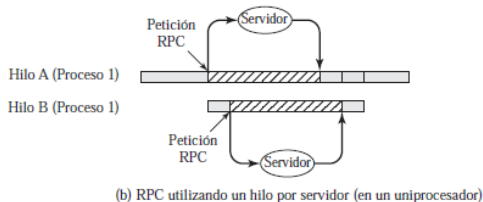
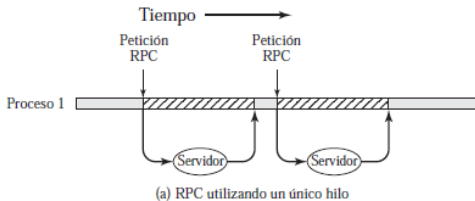
- Los principales estados de los hilos son Ejecutando, Listo y Bloqueado.
- El estado Suspendido se aplica solo a nivel proceso ya que si se expulsa un proceso de memoria, todos sus hilos deben también ser expulsados. Porqué?

Funcionalidad de los Hilos

Estados de los hilos

- Los principales estados de los hilos son Ejecutando, Listo y Bloqueado.
- El estado Suspendido se aplica solo a nivel proceso ya que si se expulsa un proceso de memoria, todos sus hilos deben también ser expulsados. Porqué? porque comparten el mismo espacio de direcciones.

Monohilo vs Multihilo



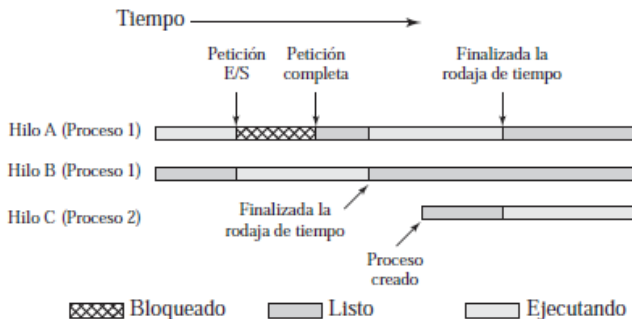
▨ Bloqueado, esperando respuesta RPC

□ Bloqueado, esperando el procesador, que está en uso por Hilo B

□ Ejecutando

- Programa que realiza 2 RPC a dos máquinas diferentes y poder combinar resultados.
- en *a* los resultados se obtienen en secuencia.
- en *b* se mejora la velocidad de ejecución del programa.

Multihilo en uniprocador con multiprogramación



- Multiprogramación permite **intercalado** de hilos de varios procesos pero nunca se llegan a ejecutar en paralelo.

Sincronización de hilos

- Debido a que todos los hilos de un proceso comparten recursos y espacio de direcciones, cualquier modificación de un recurso (archivo) por parte de un hilo, afecta el entorno de todos los hilos del mismo proceso.
- Por este motivo es necesario sincronizar las actividades de los hilos para que no interfieran entre ellos o corrompan estructuras de datos.
- Estudiaremos la sincronización de procesos e hilos más adelante.

Contenido I

1 Creación y Terminación de procesos

2 Estados

- Modelos de estados de procesos
- Swapping

3 Procesos e hilos

- Funcionalidad de los Hilos
- Hilos de nivel de usuario y de nivel núcleo

4 Procesos en Windows y Ubuntu

Hilos de nivel de usuario y de nivel núcleo

- **Hilos de nivel usuario (ULT):** la aplicación gestiona todo el trabajo de los hilos y el núcleo no es consciente de la existencia de los mismos. Se utilizan bibliotecas de hilos para gestionarlos (creación, suspensión, destrucción).
- **Hilos de nivel núcleo (KLT):** El núcleo realiza todo el trabajo de gestión de hilos. No hay código de gestión de hilos en la aplicación, solamente una API para acceder a las utilidades de hilos del núcleo. Windows utiliza este enfoque.

Contenido I

1 Creación y Terminación de procesos

2 Estados

- Modelos de estados de procesos
- Swapping

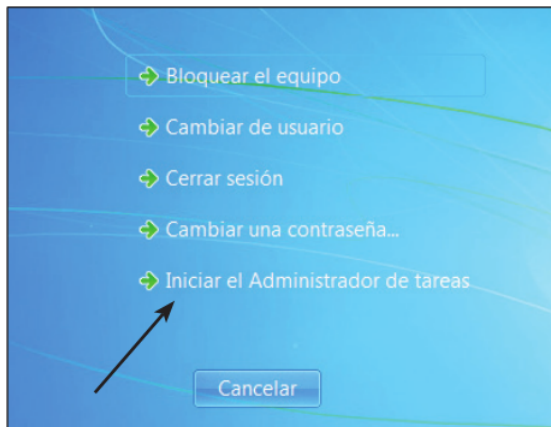
3 Procesos e hilos

- Funcionalidad de los Hilos
- Hilos de nivel de usuario y de nivel núcleo

4 Procesos en Windows y Ubuntu

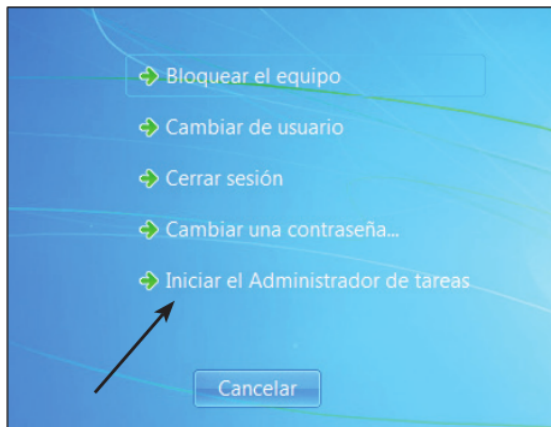
Acceder a procesos en Windows 7

- ¿Cómo accedemos a ver los procesos ejecutándose en Windows 7?



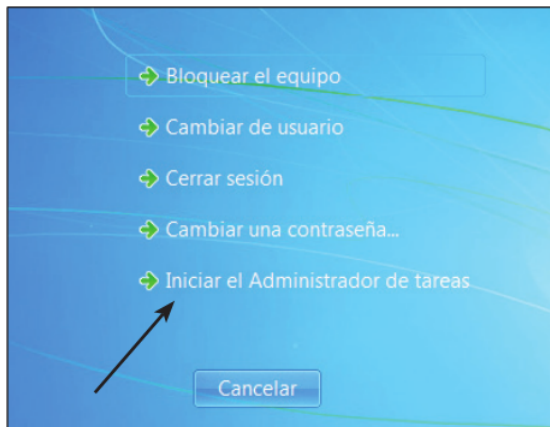
Acceder a procesos en Windows 7

- ¿Cómo accedemos a ver los procesos ejecutándose en Windows 7?
ctrl + alt + sup **Administrador de tareas**.



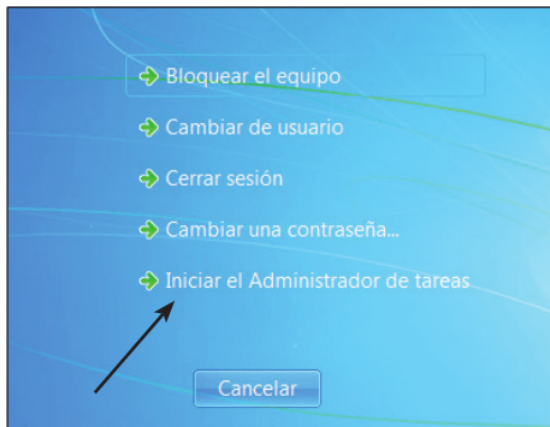
Acceder a procesos en Windows 7

- ¿Cómo accedemos a ver los procesos ejecutándose en Windows 7?
ctrl + alt + supr **Administrador de tareas**.
- ¿Otras opciones en Windows 7?



Acceder a procesos en Windows 7

- ¿Cómo accedemos a ver los procesos ejecutándose en Windows 7?
ctrl + alt + supr **Administrador de tareas**.
- ¿Otras opciones en Windows 7?
- ¿Opciones en Windows 8?



Procesos en Windows

Administrador de tareas de Windows

Archivo Opciones Ver Ayuda

Aplicaciones **Procesos** Servicios Rendimiento Funciones de red Usuarios

Nombre de imagen	Nombre de usuario	CPU	Memoria ...	Descripción
firefox.exe	yo mismo	02	276.556 KB	Firefox
AcroRd32.exe	yo mismo	00	93.956 KB	Adobe Reader
chrome.exe	yo mismo	00	38.688 KB	Google Chrome
chrome.exe	yo mismo	00	36.980 KB	Google Chrome
wlmail.exe	yo mismo	00	34.076 KB	Windows Live Mail
explorer.exe	yo mismo	00	32.392 KB	Explorador de Windows
plugin-container.exe	yo mismo	00	29.764 KB	Plugin Container for Firefox
WINWORD.EXE	yo mismo	05	18.732 KB	Microsoft Office Word
wlcomm.exe	yo mismo	00	14.684 KB	Windows Live Communications Platform
soffice.bin	yo mismo	00	7.228 KB	OpenOffice.org 3.3
mspaint.exe	yo mismo	00	5.212 KB	Paint
sidebar.exe	yo mismo	00	4.836 KB	Gadgets de escritorio de Windows
AcroRd32.exe	yo mismo	00	2.732 KB	Adobe Reader
msnmsgr.exe	yo mismo	00	1.604 KB	Windows Live Messenger
taskmgr.exe	yo mismo	16	1.560 KB	Administrador de tareas de Windows
taskhost.exe	yo mismo	00	1.236 KB	Proceso de host para tareas de Windows
SweetIM.exe	yo mismo	00	1.232 KB	SweetIM Instant Messenger Enhancer
jucheck.exe	yo mismo	00	1.136 KB	Java(TM) Update Checker
mssecos.exe	yo mismo	00	1.052 KB	Microsoft Security Client User Interface
csrss.exe	SYSTEM	06	992 KB	Proceso en tiempo de ejecución del cliente-servidor
wuauclt.exe	yo mismo	00	620 KB	Windows Update
winlogon.exe	SYSTEM	00	620 KB	Aplicación de inicio de sesión de Windows
jucheck.exe	yo mismo	00	604 KB	Java(TM) Update Scheduler

☐ Mostrar procesos de todos los usuarios

Finalizar proceso

Procesos: 64 Uso de CPU: 30% Memoria física: 58%

- La pestaña de procesos nos mostrará:

- La pestaña de procesos nos mostrará:
 - los procesos en ejecución.

- La pestaña de procesos nos mostrará:
 - los procesos en ejecución.
 - el usuario propietario

- La pestaña de procesos nos mostrará:
 - los procesos en ejecución.
 - el usuario propietario
 - el trabajo del procesador

- La pestaña de procesos nos mostrará:
 - los procesos en ejecución.
 - el usuario propietario
 - el trabajo del procesador
 - la cantidad de memoria

- La pestaña de procesos nos mostrará:
 - los procesos en ejecución.
 - el usuario propietario
 - el trabajo del procesador
 - la cantidad de memoria
 - descripción

- La pestaña de procesos nos mostrará:
 - los procesos en ejecución.
 - el usuario propietario
 - el trabajo del procesador
 - la cantidad de memoria
 - descripción
 - clic derecho en la cabecera permite seleccionar que aspectos mostrar.

- La pestaña de procesos nos mostrará:
 - los procesos en ejecución.
 - el usuario propietario
 - el trabajo del procesador
 - la cantidad de memoria
 - descripción
 - clic derecho en la cabecera permite seleccionar que aspectos mostrar.
 - Total de procesos activos (inferior izquierda)

- La pestaña de procesos nos mostrará:
 - los procesos en ejecución.
 - el usuario propietario
 - el trabajo del procesador
 - la cantidad de memoria
 - descripción
 - clic derecho en la cabecera permite seleccionar que aspectos mostrar.
 - Total de procesos activos (inferior izquierda)
 - clic derecho sobre proceso mostrará:

- La pestaña de procesos nos mostrará:
 - los procesos en ejecución.
 - el usuario propietario
 - el trabajo del procesador
 - la cantidad de memoria
 - descripción
 - clic derecho en la cabecera permite seleccionar que aspectos mostrar.
 - Total de procesos activos (inferior izquierda)
 - clic derecho sobre proceso mostrará:
 - Terminar proceso

- La pestaña de procesos nos mostrará:
 - los procesos en ejecución.
 - el usuario propietario
 - el trabajo del procesador
 - la cantidad de memoria
 - descripción
 - clic derecho en la cabecera permite seleccionar que aspectos mostrar.
 - Total de procesos activos (inferior izquierda)
 - clic derecho sobre proceso mostrará:
 - Terminar proceso
 - Establecer prioridad

- La pestaña de procesos nos mostrará:
 - los procesos en ejecución.
 - el usuario propietario
 - el trabajo del procesador
 - la cantidad de memoria
 - descripción
 - clic derecho en la cabecera permite seleccionar que aspectos mostrar.
 - Total de procesos activos (inferior izquierda)
 - clic derecho sobre proceso mostrará:
 - Terminar proceso
 - Establecer prioridad
 - Buscar en línea

- La pestaña de procesos nos mostrará:
 - los procesos en ejecución.
 - el usuario propietario
 - el trabajo del procesador
 - la cantidad de memoria
 - descripción
 - clic derecho en la cabecera permite seleccionar que aspectos mostrar.
 - Total de procesos activos (inferior izquierda)
 - clic derecho sobre proceso mostrará:
 - Terminar proceso
 - Establecer prioridad
 - Buscar en línea
 - Propiedades

- ¿Cómo abrimos el administrador de procesos en Ubuntu?

- ¿Cómo abrimos el administrador de procesos en Ubuntu? Varias opciones: Información similar que en Win.

- ¿Cómo abrimos el administrador de procesos en Ubuntu? Varias opciones: Información similar que en Win.
 - Buscar: Monitor del sistema

- ¿Cómo abrimos el administrador de procesos en Ubuntu? Varias opciones: Información similar que en Win.
 - Buscar: Monitor del sistema
 - Buscar/Icono de aplicaciones/monitor del sistema

Procesos en Ubuntu

- ¿Cómo abrimos el administrador de procesos en Ubuntu? Varias opciones: Información similar que en Win.
 - Buscar: Monitor del sistema
 - Buscar/Icono de aplicaciones/monitor del sistema
 - Instalar “indicador de carga del sistema”



Nombre del proceso	Usuario	% CPU	ID	Memoria	Prioridad
compiz	chalo	22	2006	170,3 MiB	Normal
gnome-system-monitor	chalo	10	3706	13,2 MiB	Normal
indicator-multiloader	chalo	1	3677	6,1 MiB	Normal
unity-scope-loader	chalo	0	3648	4,9 MiB	Normal
unity_picasa_daemon.py	chalo	0	3594	7,5 MiB	Normal
unity_shotwell_daemon.py	chalo	0	3593	5,2 MiB	Normal
unity_facebook_daemon.py	chalo	0	3590	7,1 MiB	Normal
unity_flickr_daemon.py	chalo	0	3589	7,4 MiB	Normal
unity-video-lens-daemon	chalo	0	3575	2,8 MiB	Normal
bash	chalo	0	3392	1,6 MiB	Normal
ubuntu-geoip-provider	chalo	0	2621	1,1 MiB	Normal
geoclue-master	chalo	0	2617	920,0 KiB	Normal
gvfsd-metadata	chalo	0	2570	392,0 KiB	Normal
deia-dup-monitor	chalo	0	2430	3,1 MiB	Normal