

Tema 2 Procesos

3º ITSI

Informática

tema 2
procesos

Contenido

Contenido

concepto
estados
PCB
operaciones
threads
ejemplos

Tema 2. Procesos

2.1 Concepto de proceso

2.2 Estados de un proceso

Transiciones de estado

2.3 Bloque de control de proceso (PCB)

2.4 Operaciones con procesos

2.5 Threads

2.6 Ejemplos:

Descripción de procesos en UNIX, Linux y Windows XP

Contenido**concepto**

estados

PCB

operaciones

threads

ejemplos

Dept. Arquitectura
de Computadores

3 Univ. Málaga

2.1 Concepto de proceso

*¿qué es un programa?***Contenido****concepto**

estados

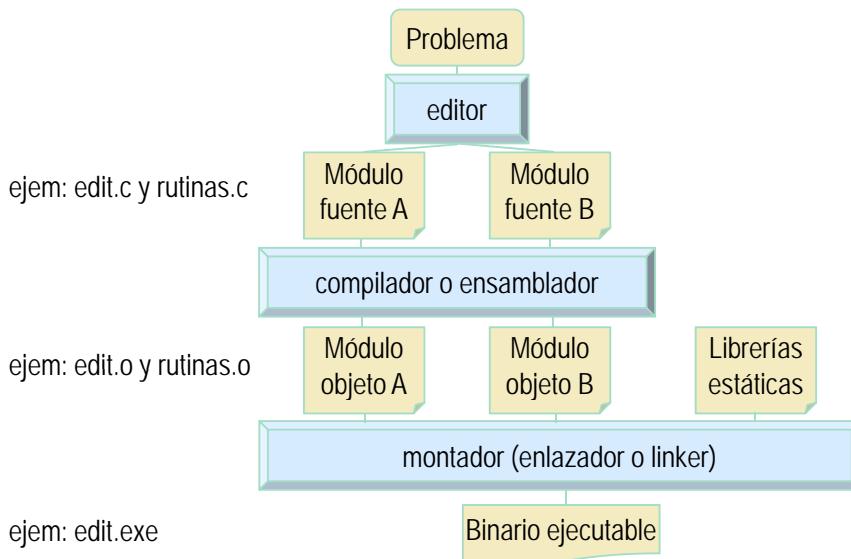
PCB

operaciones

threads

ejemplos

- ④ Secuencia de instr. que implementan un algoritmo (un problema a resolver)
- ④ Generación de un programa ejecutable:

Dept. Arquitectura
de Computadores

4 Univ. Málaga

Contenido

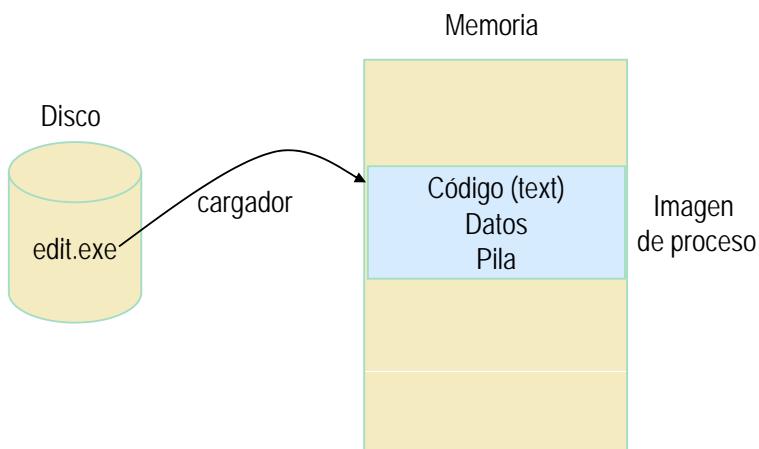
concepto
estados
PCB
operaciones
threads
ejemplos

Dept. Arquitectura
de Computadores

5 Univ. Málaga

Ejecución de un programa

- Una instancia del programa: "C: > edit <enter>"
- Un programa del SO (el cargador) busca espacio en Mem
- Carga el código y los datos. Reserva espacio para la pila

**Contenido**

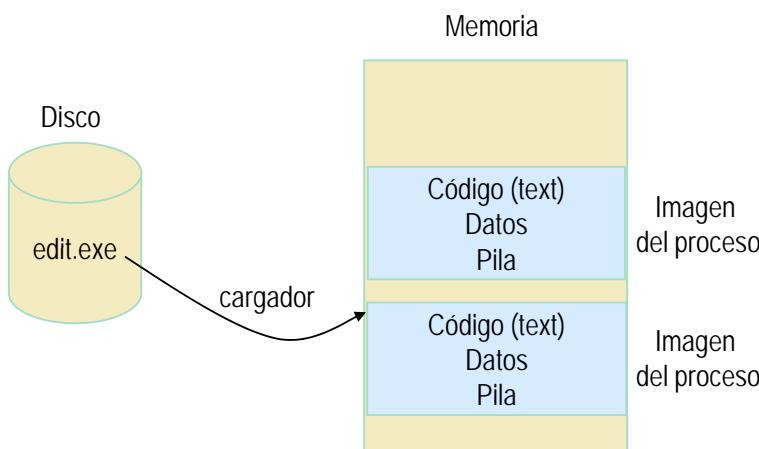
concepto
estados
PCB
operaciones
threads
ejemplos

Dept. Arquitectura
de Computadores

6 Univ. Málaga

Ejecución de un programa

- Otra instancia del programa: "C: > edit <enter>"
- Necesitas una región de datos y pila diferentes
- Los SO antiguos también replican el código
 - Pero no es necesario si éste no se modifica durante la ejecución



Contenido

concepto
estados
PCB
operaciones
threads
ejemplos

Dept. Arquitectura
de Computadores

7 Univ. Málaga

④ Definiciones:

- Programa en ejecución.
- Instanciación de un programa

④ Otras definiciones:

- Elemento SW susceptible de ser planificado
- Actividad asincrónica
- “Espíritu animado del procedimiento” 😊
Operating Systems. H.M. Deitel. Addison Wesley. 1990.

Contenido

concepto
estados
PCB
operaciones
threads
ejemplos

Dept. Arquitectura
de Computadores

8 Univ. Málaga

2.2 Estados de un proceso

Contenidoconcepto
estadosPCB
operaciones
threads
ejemplosDept. Arquitectura
de Computadores

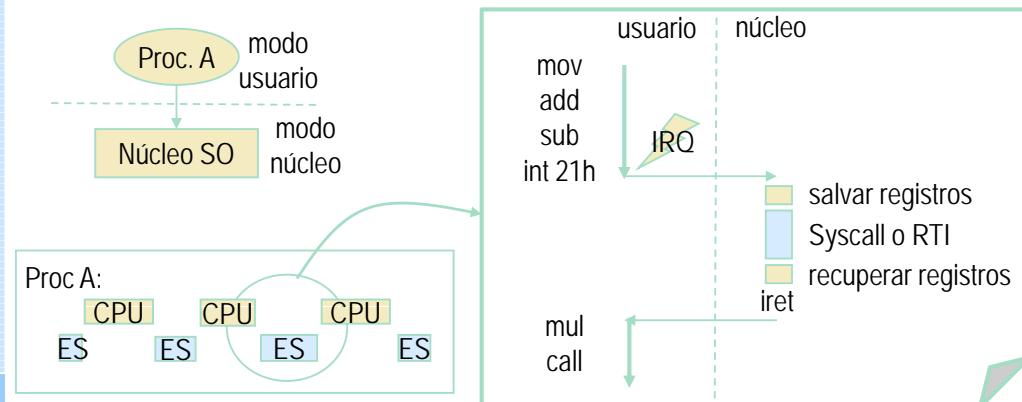
9 Univ. Málaga

c) Monotarea:

- Sólo cuando acaba un proceso puede ejecutarse otro



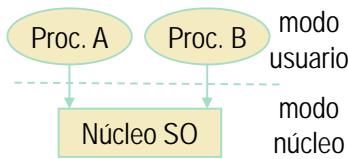
- El proceso tiene ráfagas de CPU y de E/S
 - Las operaciones de E/S las "sirve" el SO mediante syscalls
 - Durante la ejecución del proc. pueden llegar IRQs que atiende el SO

**Contenido**concepto
estadosPCB
operaciones
threads
ejemplosDept. Arquitectura
de Computadores

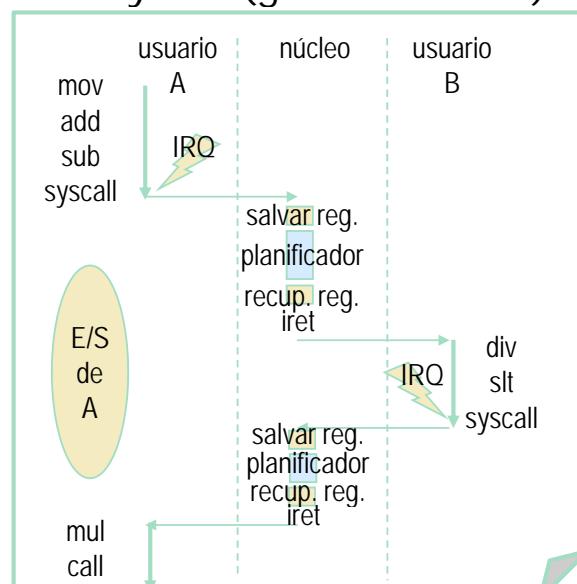
10 Univ. Málaga

c) Multiprogramación y Multitarea

- La memoria almacena varios procesos
 - Grado de multiprogramación: número de procesos activos
- Paralelismo entre CPU y E/S (gracias al DMA)



- Cambio de contexto
 - Iniciado por
 - Syscall (Int SW)
 - » E/S
 - » exit()
 - IRQ (Int HW)
 - » E/S, reloj

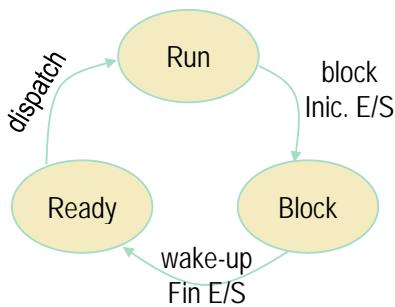


Contenidoconcepto
estadosPCB
operaciones
threads
ejemplosDept. Arquitectura
de Computadores

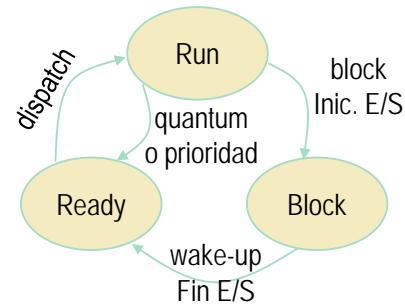
11 Univ. Málaga

E Estados

- Ejecución (un proceso por procesador)
- Ready (todos los procesos listos para ejecutarse)
- Blocked (todos los procesos esperando evento)

**Sistema multiprogramado**

Hasta que el proceso en ejecución no llama al SO no puede despacharse ningún otro proceso

**Sistema multitarea**

Aunque el proceso en ejecución no llame al SO, otro proceso puede despacharse

Contenidoconcepto
estadosPCB
operaciones
threads
ejemplosDept. Arquitectura
de Computadores

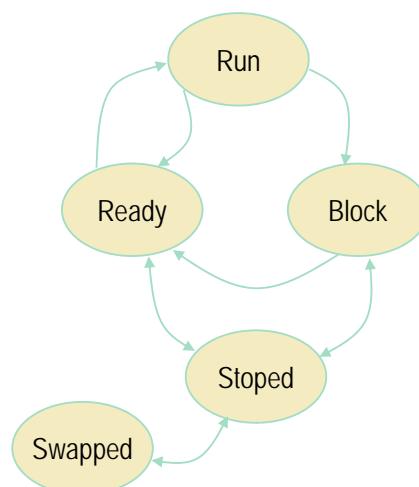
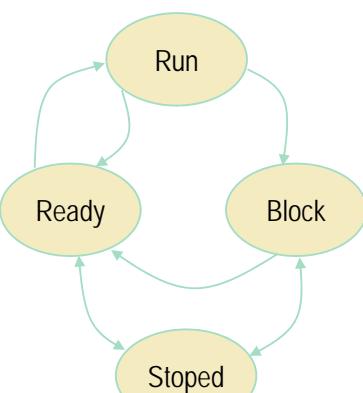
12 Univ. Málaga

E Estado parado o suspendido (stoped): útil en

- Sistema cargado, depuración, proceso sospechoso, demonios...

E Estado descargado (swapped):

- Liberar memoria de un proceso parado (se guarda en HD)



2.2 Estados de un proceso

un modelo de 5 estados

Contenido

concepto
estados

PCB
operaciones
threads
ejemplos

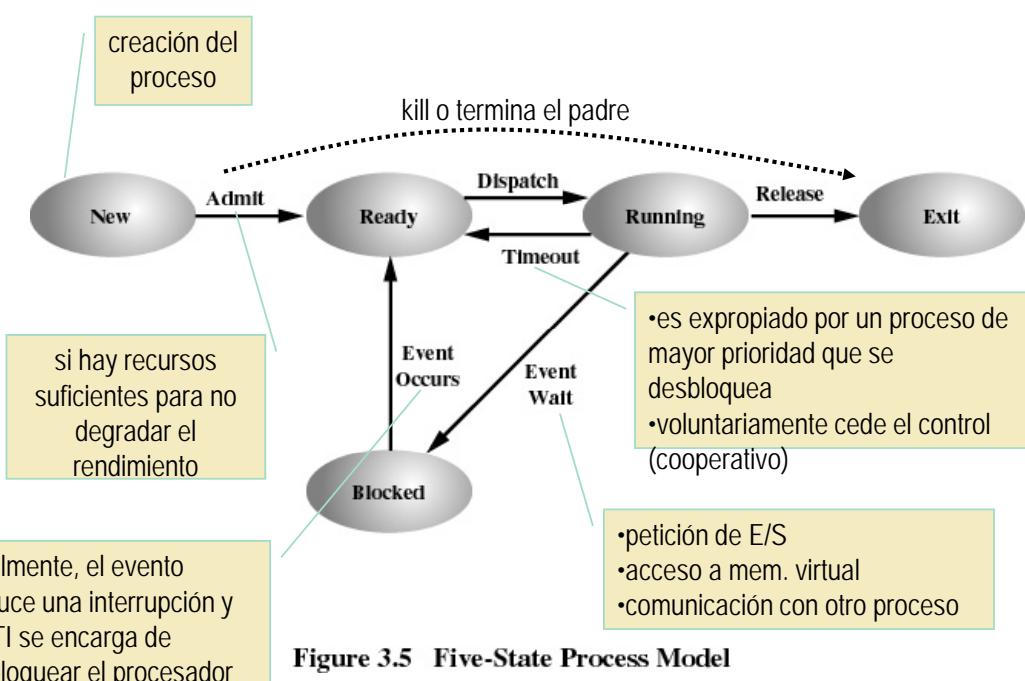


Figure 3.5 Five-State Process Model

Dept. Arquitectura
de Computadores

13 Univ. Málaga

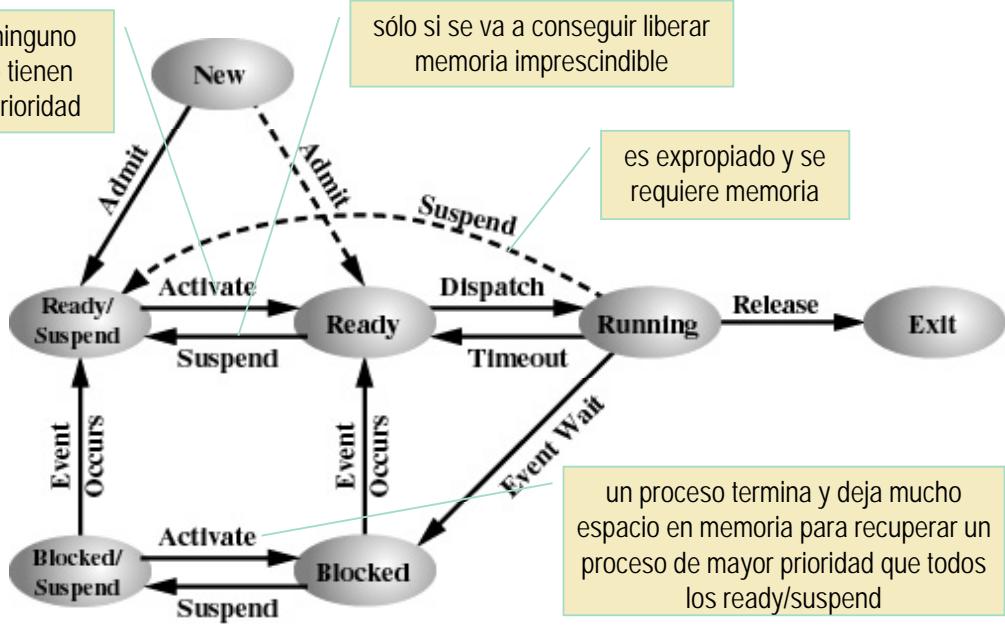
2.2 Estados de un proceso

diagrama con dos estados de suspendido

Contenido

concepto
estados

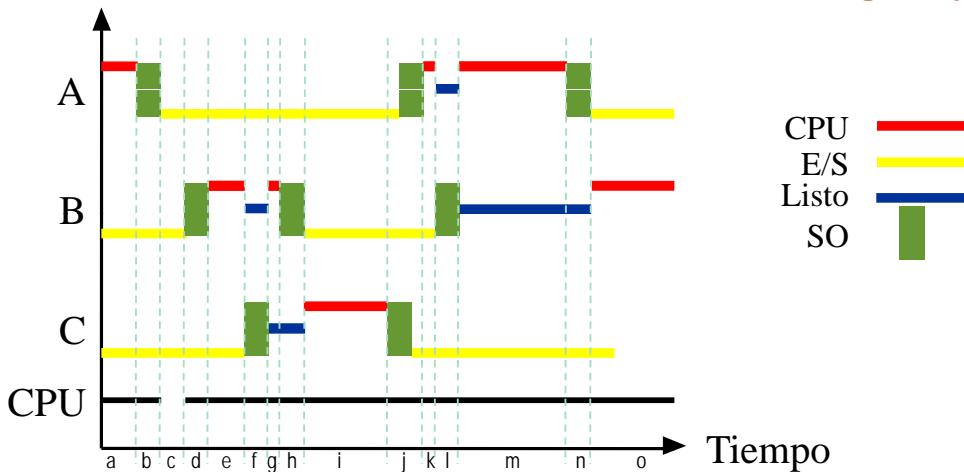
operaciones
threads
ejemplos



(b) With Two Suspend States

Dept. Arquitectura
de Computadores

14 Univ. Málaga

Contenidoconcepto
estadosPCB
operaciones
threads
ejemplos

Fuente: Sistemas Operativos. Una visión aplicada.

- a: A en CPU, B y C bloqueados
 b: A llama al SO para E/S
 c: Todos bloqueados (CPU idle)
 d: Acaba E/S de B (despierta y dispatch)
 e: B en ejecución
 f: Acaba E/S de C (despierta), B ready
 g: Sigue B en CPU y C ready
 h: B hace syscall. SO despacha C y bloquea B

- i: C en CPU, A y B bloqueados
 j: C llama al SO para E/S y A se despierta
 k: A en ejecución
 l: Una int E/S llama al SO para despertar B
 m: A continua en Run y B espera ready
 n: A se bloquea
 o: B pasa a ejecución (run)
CPU ocupada siempre salvo en c

Dept. Arquitectura
de Computadores

15 Univ. Málaga

Contenidoconcepto
estadosPCB
operaciones
threads
ejemplosDept. Arquitectura
de Computadores

16 Univ. Málaga

Ventajas de la multitarea

- Aprovecha el tiempo de espera de un proceso mientras termina una operación de E/S
- Aumenta el uso de CPU
- Permite una implementación eficiente de sistemas multiusuario
- Facilita la programación
 - Divide un programa en procesos que colaboran entre sí
 - Modularidad en la programación

Limitación del grado de multiprogr.

- Si hay muchos procesos ready aumenta el tiempo de respuesta
- El tamaño limitado de la memoria también restringe el número de procesos cargados

Contenido

concepto

estados

PCB

operaciones

threads

ejemplos

Dept. Arquitectura
de Computadores

17 Univ. Málaga

2.3 Bloque de control de proc.**Contenido**

concepto

estados

PCB

operaciones

threads

ejemplos

Dept. Arquitectura
de Computadores

18 Univ. Málaga

- ④ El S.O. necesita estructuras para mantener:
- ④ Información sobre la **memoria** (asignada, atributos, VM, ...)
- ④ Información sobre los **dispositivos E/S** (asignación, estado, ...)
- ④ Información sobre los **ficheros** (asignación, estado, ...)
- ④ Información sobre los **procesos** (estado, control, memoria, ...)

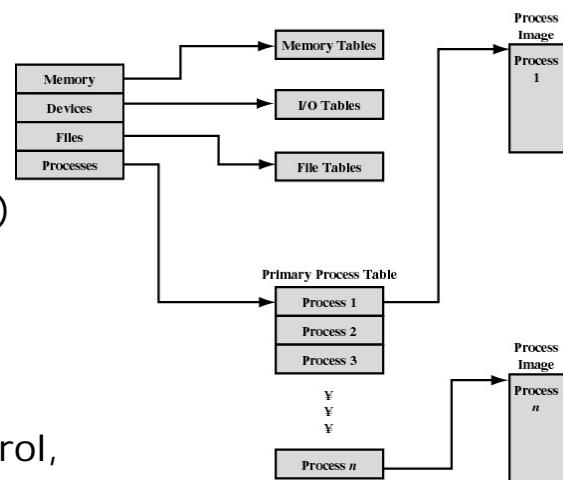


Figure 3.10 General Structure of Operating System Control Tables

Contenidoconcepto
estados

PCB

operaciones
threads
ejemplosDept. Arquitectura
de Computadores

19 Univ. Málaga

④ Información de un proceso

- **Estado del procesador**
 - Contenido de los registros del procesador
- **Imagen de memoria (core image):**
 - contenido de los segmentos de memoria de código, datos y pila
- **Información para manejar cada proceso**
 - Bloque de Control de Procesos (BCP, PCB)

Contenidoconcepto
estados

PCB

operaciones
threads
ejemplosDept. Arquitectura
de Computadores

20 Univ. Málaga

④ PCB

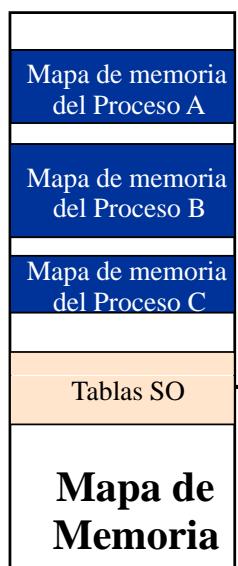
- **Identificación del proceso**
 - Id. del proceso, del padre, usuario, ...
- **Información del Estado del Procesador**
 - Registros visibles por el usuario, PC, registro estado, punteros a la Pila, ...
- **Información de Control del Proceso**
 - Información de Planificación y Estado (*Estado del Proceso, Prioridad, Información para planificación, ...*)
 - Comunicación Inter-procesos (señales y mensajes)
 - Privilegios del Proceso
 - Gestión de Memoria (punteros a tablas de páginas, ...)
 - Utilización y Propiedad de Recursos (ficheros abiertos, ...)

2.3 PCB

información de un proceso

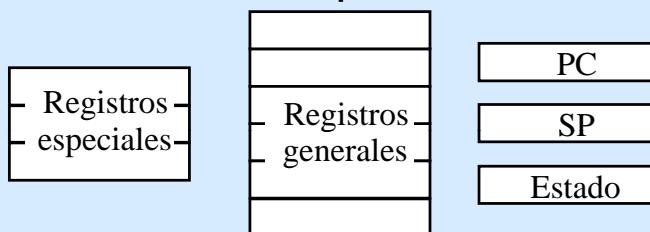
Contenido

concepto
estados
PCB
operaciones
threads
ejemplos

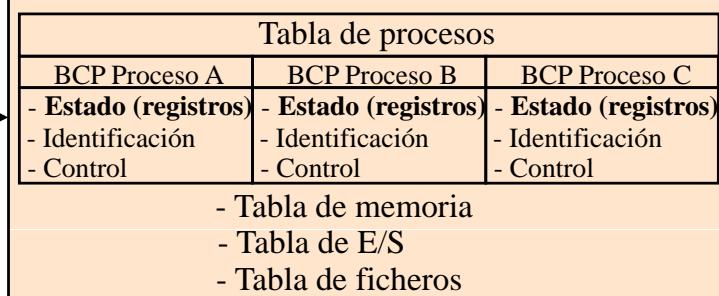


Dept. Arquitectura
de Computadores

Estado del procesador



Tablas del sistema operativo



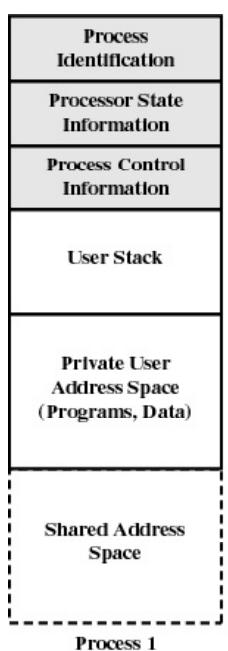
Fuente: Sistemas Operativos. Una visión aplicada.

2.3 PCB

imágenes de procesos

Contenido

concepto
estados
PCB
operaciones
threads
ejemplos



Dept. Arquitectura
de Computadores

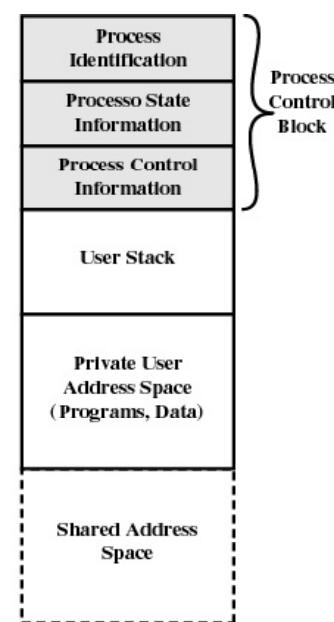
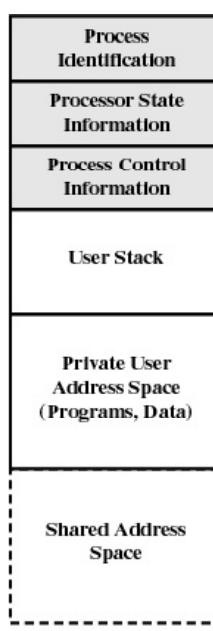


Figure 3.12 User Processes in Virtual Memory

Contenidoconcepto
estados**PCB**

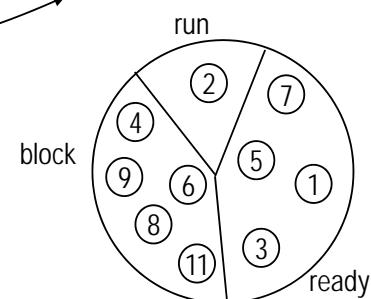
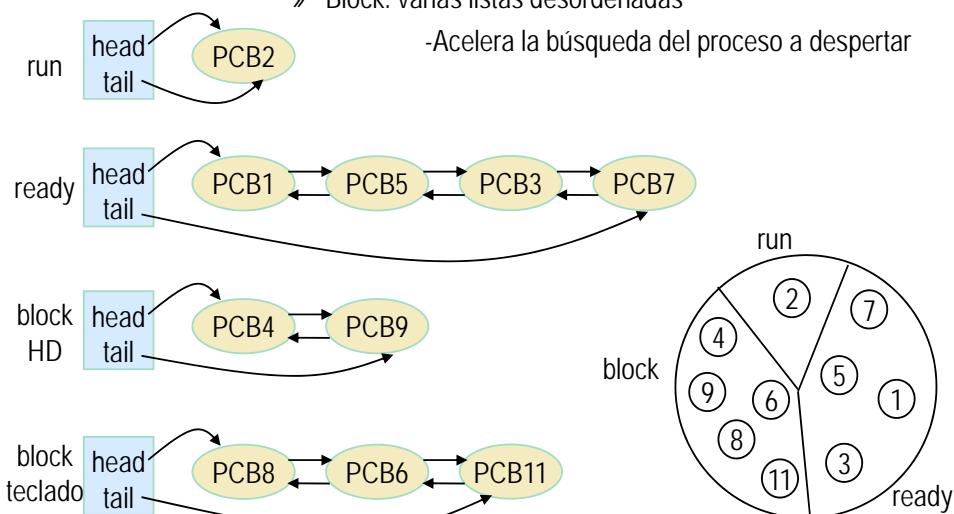
operaciones

threads

ejemplos

Dept. Arquitectura
de Computadores**Organización de los procesos****■ Listas de procesos de un tipo determinado**

- » Run: tantos procesos como procesadores
- » Ready: procesos listos. Lista ordenada por el planificador
- » Block: varias listas desordenadas

**Contenido**concepto
estados**PCB**

operaciones

threads

ejemplos

Dept. Arquitectura
de Computadores**Implementación frecuente de listas de PCB****■ Array de PCBs con índices (punteros) al siguiente.****Tabla de procesos**

PCB1	PCB2	PCB3	PCB4	PCB5	PCB6	PCB7	PCB8	PCB9	PCB10	PCB11	PCB12
5	0	7	9	3	11	0	6	0	0	0	

ready → 1 4 ← Punteros de las colas

■ Consecuencia:

- El PCB tiene que ser de tamaño fijo
- Las tablas de tamaño variable están fuera del PCB
 - Tablas de Mem, Tablas de E/S, Tablas de ficheros
 - Estas tablas son apuntadas desde el PCB

Contenidoconcepto
estados
PCB**operaciones**threads
ejemplosDept. Arquitectura
de Computadores

25 Univ. Málaga

2.4 Operaciones con proc.

Contenidoconcepto
estados
PCB**operaciones**threads
ejemplosDept. Arquitectura
de Computadores

26 Univ. Málaga

- ④ Un proceso puede crear otros procesos, y
 - Formar un árbol de procesos (UNIX): relación de parentesco entre procesos.
 - No mantener una jerarquía (Win 2000).
- ④ ¿Compartir recursos entre el creador y el creado?
 - Comparten todos los recursos, o un subconjunto.
 - Creador y creado no comparten recursos.
- ④ Respecto a la ejecución:
 - Creador/creado se ejecutan concurrentemente.
- ④ Sus espacios de direcciones son:
 - Clonados: se copia el espacio del creador para el creado. ej. Unix.
 - Nuevos: el proceso creado inicia un programa diferente al del creador. ej. Windows 2000.

Contenidoconcepto
estados
PCBoperaciones
threads
ejemplosDept. Arquitectura
de Computadores

- ➊ Un proceso finaliza cuando ejecuta la declaración de terminación (explícita o implícitamente).
- ➋ Los pasos a seguir:
 1. Envío de datos del proceso creado a creador. P.ej. Código de finalización.
 2. El SO libera los recursos que tiene.
- ➌ El proceso puede finalizar la ejecución de otro si:
 - Ha sobrepasado los recursos asignados.
 - La tarea asignada al proceso ya no es necesaria.
 - Va a finalizar y el SO no permite que los procesos creados por él puedan continuar: terminación en cascada.

Contenidoconcepto
estados
PCBoperaciones
threads
ejemplosDept. Arquitectura
de Computadores

- ➊ Operaciones con procesos
 - Identificación de procesos
 - getpid, GetCurrentProcessId
 - Entorno de un proceso
 - getenv, GetEnvironmentStrings
 - Path, directorio de trabajo, tipo de terminal, dir. personal
 - Creación de procesos
 - fork, CreateProcess
 - Cambio de programa de un proceso: exec
 - Esperar la terminación de un proceso
 - wait, WaitForSingleObject, WaitForMultipleObjects
 - Finalizar la ejecución de un proceso
 - exit, TerminateProcess
 - Enviar señales a un proceso: kill
 - Comunicar procesos

Contenido

concepto

estados

PCB

operaciones

threads

ejemplos

Operación	Unix	Win32
Crear	<code>fork()</code> <code>exec()</code>	<code>CreateProcess()</code>
Terminar	<code>_exit()</code>	<code>ExitProcess()</code>
Obtener código finalización	<code>wait</code> <code>waitpid</code>	<code>GetExitCodeProcess</code>
Obtener tiempos	<code>times</code> <code>wait3</code> <code>wait4</code>	<code>GetProcessTimes</code>
Identificador	<code>getpid</code>	<code>GetCurrentProcessId</code>
Terminar otro proceso	<code>kill</code>	<code>TerminateProcess</code>

Dept. Arquitectura
de Computadores**Contenido**

concepto

estados

PCB

operaciones

threads

ejemplos

@ **Servicios POSIX****pid_t fork(void)**

- Crea un proceso hijo. Devuelve 0 al proceso hijo y el pid del hijo al proceso padre.

int execp(const char *file, const char *arg, ...)

- Permite a un proceso ejecutar un programa (código) distinto. Cambia la imagen de memoria del proceso. El pid no cambia.

pid_t wait(int *status)

- Permite a un proceso padre esperar hasta que termine un proceso hijo. Devuelve el identificador del proceso hijo y el estado de terminación del mismo.

void exit(int status)

- Finaliza la ejecución de un proceso indicando el estado de terminación del mismo.

pid_t getpid(void)

- Devuelve el identificador del proceso.

pid_t getppid(void)

- Devuelve el identificador del proceso padre.

Dept. Arquitectura
de Computadores

Contenido

concepto

estados

PCB

operaciones

threads

ejemplos

Dept. Arquitectura
de Computadores

31 Univ. Málaga

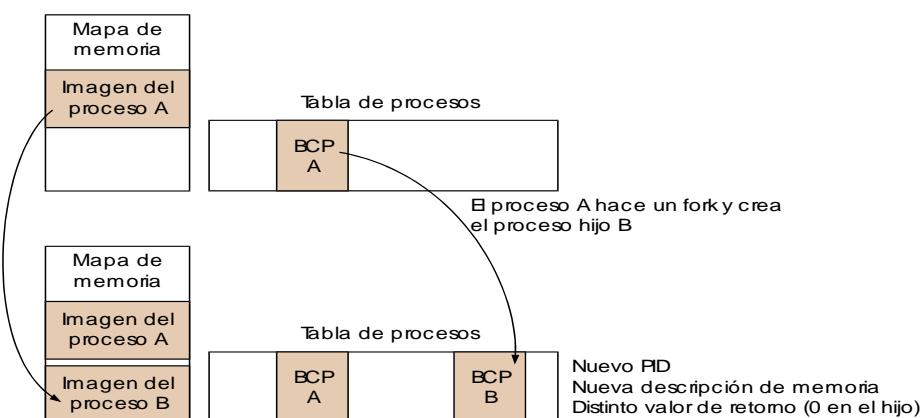
• `pid_t fork(void);`

- Devuelve:

- El identificador de proceso hijo al proceso padre y 0 al hijo
- -1 el caso de error

- Descripción:

- Crea un proceso hijo que ejecuta el mismo progr. que el padre
- Hereda los ficheros abiertos (se copian los descriptores).

**Contenido**

concepto

estados

PCB

operaciones

threads

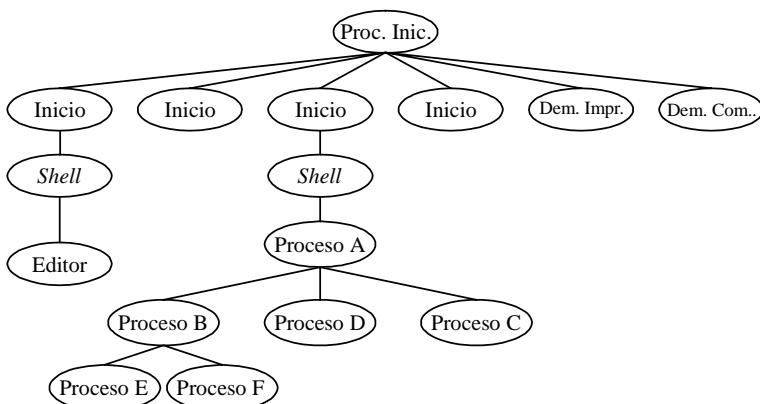
ejemplos

Dept. Arquitectura
de Computadores

32 Univ. Málaga

• Unix mantiene las relaciones padre-hijo

- El primer proceso es Init (padre del resto)
- Init lanza demonios procesos de login
- El login se convierte en shell
- Desde el shell el usuario lanza procesos



Contenido

concepto

estados

PCB

operaciones

threads

ejemplos

Dept. Arquitectura
de Computadores

④ int **execvp**(const char *file, const char *arg, ...)

- Argumentos:

- path, file: nombre del archivo ejecutable
- arg: argumentos

- Descripción:

- Devuelve -1 en caso de error, en caso contrario **no** retorna.
- Cambia la imagen de memoria del proceso.
- El mismo proceso ejecuta otro programa.
- Los ficheros abiertos permanecen abiertos

⑤ int **exit**(int status);

- Argumentos: Código de retorno al proceso padre

- Descripción:

- Finaliza la ejecución del proceso.
- Se cierran todos los descriptores de ficheros abiertos.
- Se liberan todos los recursos del proceso

Contenido

concepto

estados

PCB

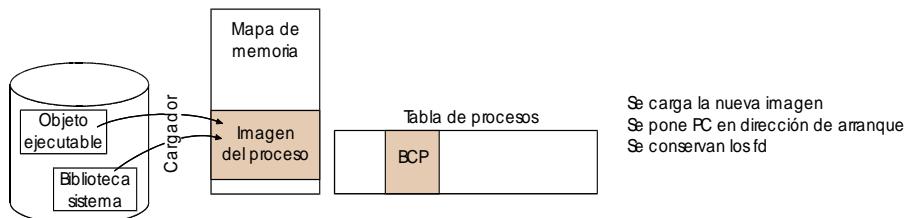
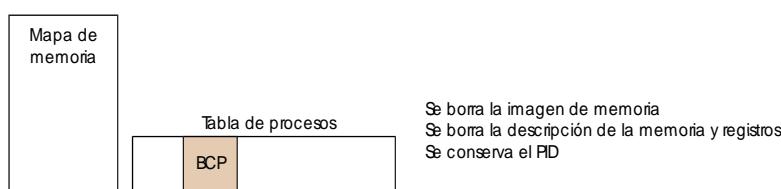
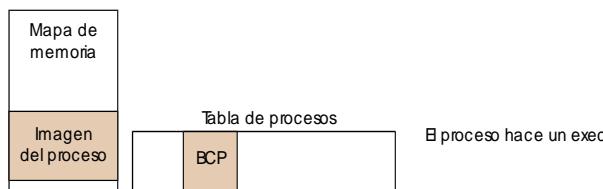
operaciones

threads

ejemplos

Dept. Arquitectura
de Computadores

⑥ exec() cambia el código de un proceso



Contenido

concepto

estados

PCB

operaciones

threads

ejemplos

Dept. Arquitectura
de Computadores

35 Univ. Málaga

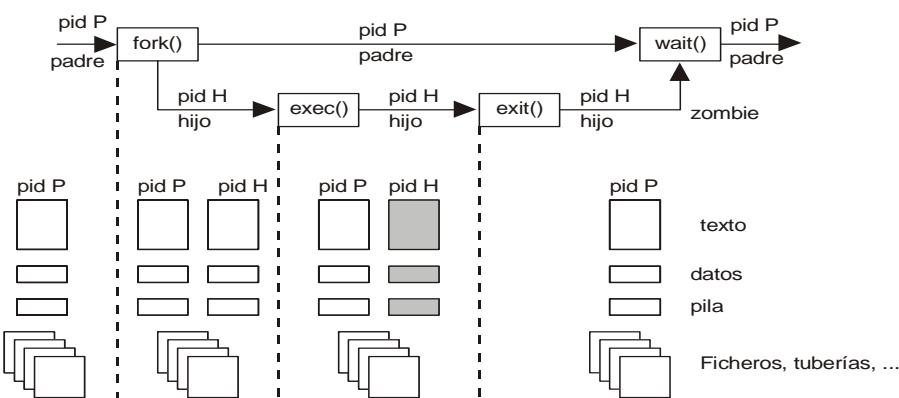
@ `pid_t wait(int *status);`

▪ Argumentos:

- Devuelve el código de terminación del proceso hijo.

▪ Descripción:

- Devuelve el identificador del proceso hijo o -1 en caso de error.
- Permite a un proceso padre esperar hasta que termine un proceso hijo. Devuelve el identificador del proceso hijo y el estado de terminación del mismo.

**Contenido**

concepto

estados

PCB

operaciones

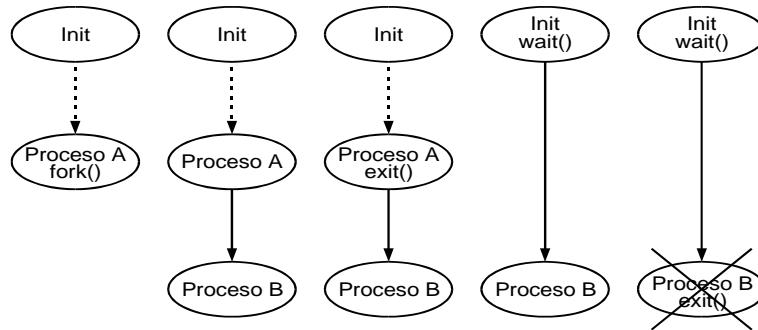
threads

ejemplos

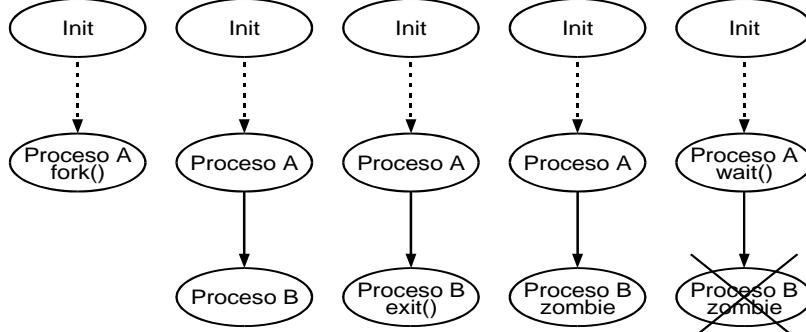
Dept. Arquitectura
de Computadores

36 Univ. Málaga

@ Si el padre muere: init "adota" a los hijos



@ Zombie: desde que el hijo hace exit hasta que el padre hace wait



Contenidoconcepto
estados

PCB

operaciones

threads

ejemplos

Dept. Arquitectura
de Computadores

@ Un proceso que lanza un "ls -l"

```
#include <sys/types.h>
#include <stdio.h>
/* programa que ejecuta el mandato ls -l */
main() {
    pid_t pidfork;
    int status;
    pidfork = fork();
    if (pidfork == 0) /* proceso hijo */
        execlp("ls","ls","-l",NULL);
        exit(-1);
    }
    else /* proceso padre */
        while (pidfork != wait(&status));
    exit(0);
}
```

Contenidoconcepto
estados

PCB

operaciones

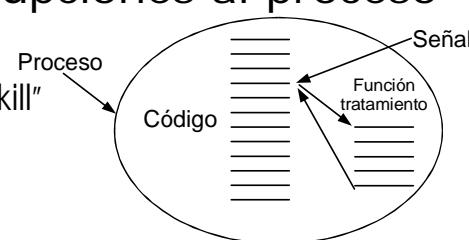
threads

ejemplos

Dept. Arquitectura
de Computadores

@ Las señales son interrupciones al proceso

- Se envían
 - de proceso a proceso con "kill"
 - del SO a un proceso
- Tipos de señales ("kill -l" da la lista)
 - SIGKILL -9- (mata un proceso: no enmascarable)
 - SIGTERM -15- (termina un proceso: enmascarable)
 - SIGSTOP -19- (suspende un proceso)
 - SIGCONT (reanuda un proceso)
- El proceso debe estar preparado para recibir la señal (especifica un proced.) o la enmascara
 - Si no está preparado: muere o la ignora



Fuente: Sistemas Operativos. Una visión aplicada.

2.4 Operaciones con proc. servicios para gestión de señales

Contenido

concepto

estados

PCB

operaciones

threads

ejemplos

Dept. Arquitectura
de Computadores

- **int kill(pid_t pid, int sig)**
 - Envía al proceso "pid" la señal "sig"
- **int sigaction(int sig, struct sigaction *act, struct sigaction *oact)**
 - Permite especificar la acción a realizar como tratamiento de la señal "sig"
- **int pause(void)**
 - Bloquea al proceso hasta la recepción de una señal.
- **unsigned int alarm(unsigned int seconds)**
 - Genera la recepción de la señal SIGALRM pasados "seconds" segundos.
- **sigprocmask(int how, const sigset_t *set, sigset_t *oset)**
 - Se utiliza para examinar o modificar la máscara de señales de un proceso.

2.4 Operaciones con proc. procesos demonios

Contenido

concepto

estados

PCB

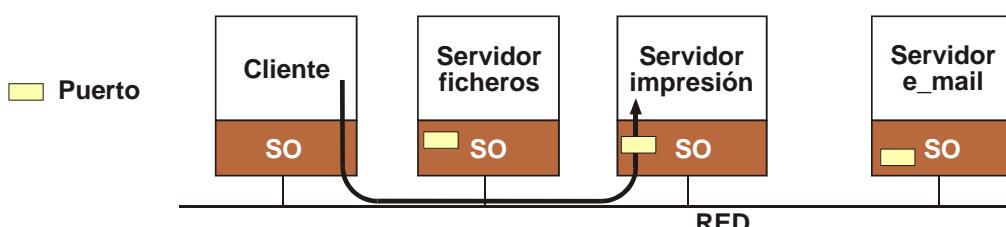
operaciones

threads

ejemplos

Dept. Arquitectura
de Computadores

- Es un proceso que ejecuta:
 - En background (su padre no le espera)
 - No asociado a un terminal o proceso login
 - Que espera que ocurra un evento (petición de un cliente)
 - O que debe realizar una tarea de forma periódica
- Características
 - Se arrancan al iniciar el sistema
 - No mueren
 - Están normalmente en espera de evento
 - No hacen el trabajo, lanzan otros procesos o threads



Contenidoconcepto
estados
PCB**operaciones**threads
ejemplosDept. Arquitectura
de Computadores

@ Servicios Win32:

- Crear un proceso
 - `BOOL CreateProcess(....);`
- Terminar la ejecución de un proceso
 - `VOID ExitProcess(UINT nExitCode);`
- Obtener el código de term. de un proceso
 - `BOOL GetExitCodeProcess(HANDLE hProcess, LPDWORD lpdwExitCode);`
- Finalizar la ejecución de otro proceso
 - `BOOL TerminateProcess(HANDLE hProcess, UINT uExitCode);`
- Esperar por la finalización de un proceso
 - `DWORD WaitForSingleObject(HANDLE hObject, DWORD dwTimeOut);`
 - `DWORD WaitForMultipleObjects(DWORD cObjects, LPHANDLE lphObjects, BOOL fWaitAll, DWORD dwTimeOut);`

Contenidoconcepto
estados
PCB**operaciones**threads
ejemplosDept. Arquitectura
de Computadores

2.5 Threads

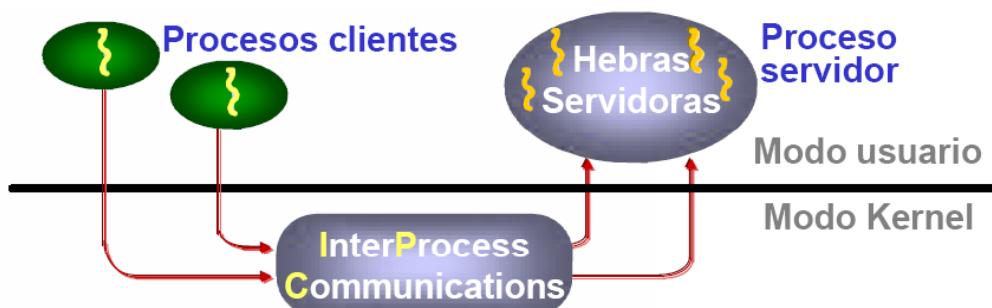
Contenido

concepto
estados
PCB
operaciones
threads
ejemplos

Dept. Arquitectura
de Computadores

43 Univ. Málaga

- ➊ Hay aplicaciones por naturaleza paralelas que necesitan un espacio común de direcciones.
 - Servidores de bases de datos
 - Monitores de procesamiento de transacciones
 - Procesamiento de protocolos de red
 - Etc.

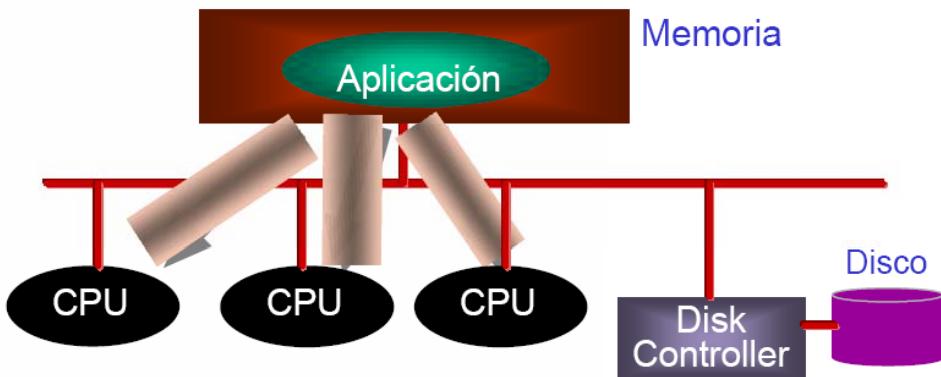
**Contenido**

concepto
estados
PCB
operaciones
threads
ejemplos

Dept. Arquitectura
de Computadores

44 Univ. Málaga

- ➋ Los procesos no pueden sacar partido de las arquitecturas multi-procesador
 - Un proceso sólo puede usar un procesador a la vez



Contenido

concepto
estados
PCB
operaciones
threads
ejemplos

Dept. Arquitectura
de Computadores

45 Univ. Málaga

- ➊ Aspectos separables que confluyen en un proceso:
 - **Unidad planificable (flujo de control)**
 - Secuencia de instrucciones a ejecutar; determinadas por el PC, la pila y los registros.
 - **Unidad propietaria de los recursos**
 - Direcciones de memoria a las que se puede acceder y los recursos asignados (archivos, E/S, ...)
- ➋ Queremos introducir un modelo de ejecución diferente del modelo de proceso visto pero que sea compatible

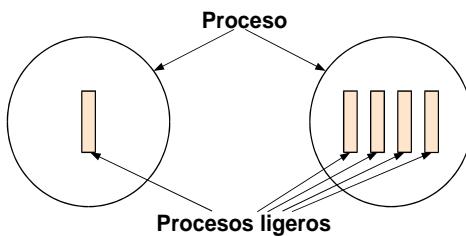
Contenido

concepto
estados
PCB
operaciones
threads
ejemplos

Dept. Arquitectura
de Computadores

46 Univ. Málaga

- ➌ Idea: **Permitir más de un flujo de control dentro del mismo espacio de direcciones**
 - Dentro del mismo programa (datos y código) y los mismos recursos del SO, permitimos varias ejecuciones.
- ➍ Separar las 2 unidades de un proceso
 - Proceso: sólo la unidad propietaria de recursos
 - Thread (hilo): sólo la unidad planificable
- ➎ Posibilidad: varios threads en un proceso



Contenido

concepto
estados
PCB
operaciones
threads
ejemplos

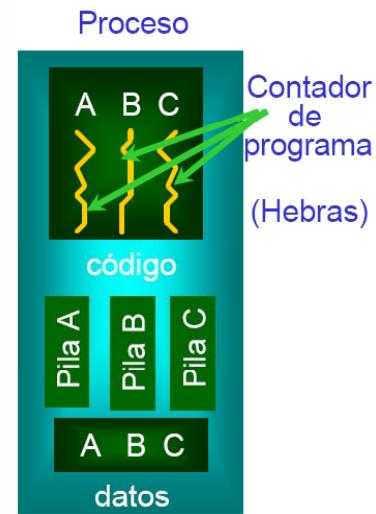
Dept. Arquitectura
de Computadores

47 Univ. Málaga

- ➊ **Thread/Hebra:** Unidad de asignación de la CPU (de planificación)

- ➋ Tiene su propio contexto hardware

- Su valor del contador de programa
- Los valores de los registros
- Su pila de ejecución

**Contenido**

concepto
estados
PCB
operaciones
threads
ejemplos

Dept. Arquitectura
de Computadores

48 Univ. Málaga

- ➊ Información propia del thread

- Estado de ejecución del thread (run, ready, block)
- Contexto del thread (guardado cuando no se está ejecutando): PC, registros.
- Pila
- Almacenamiento estático per-thread para variables locales

- ➋ Todos los threads comparten el acceso a la memoria y recursos de su proceso:

- Espacio de memoria
- Variables globales
- Archivos abiertos
- Procesos hijos
- Señales

Contenido

concepto
estados
PCB
operaciones
threads
ejemplos

Dept. Arquitectura
de Computadores

49 Univ. Málaga

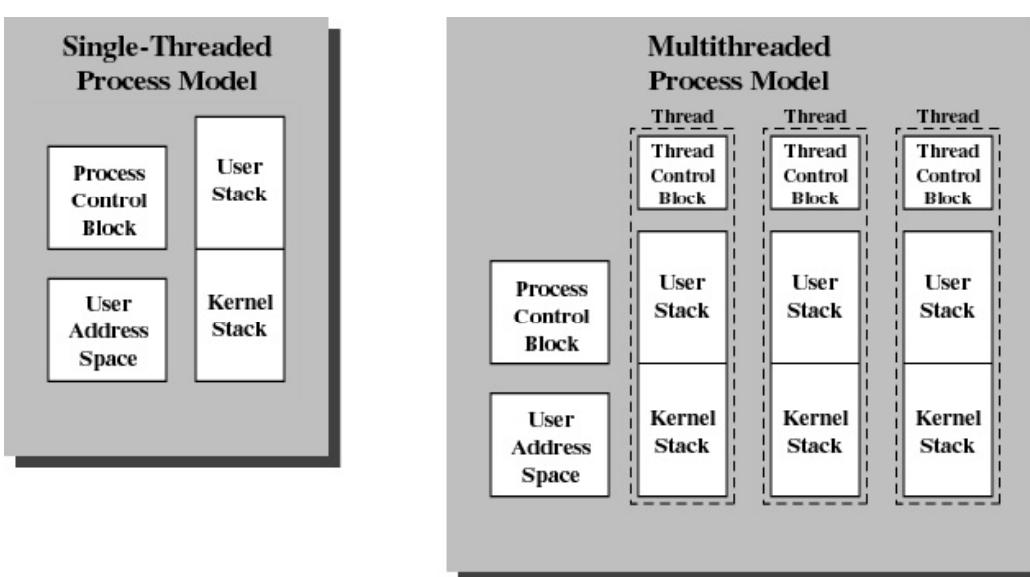


Figure 4.2 Single Threaded and Multithreaded Process Models

Contenido

concepto
estados
PCB
operaciones
threads
ejemplos

Dept. Arquitectura
de Computadores

50 Univ. Málaga

- ➊ El sistema operativo permite múltiples threads de ejecución por cada proceso
- ➋ MS-DOS sólo tiene un thread
- ➌ UNIX permite múltiples procesos de usuario, pero sólo un thread por proceso
- ➍ Windows XP, Solaris y Linux soportan múltiples threads.

Contenido

concepto
estados
PCB
operaciones
threads
ejemplos

Dept. Arquitectura
de Computadores

51 Univ. Málaga

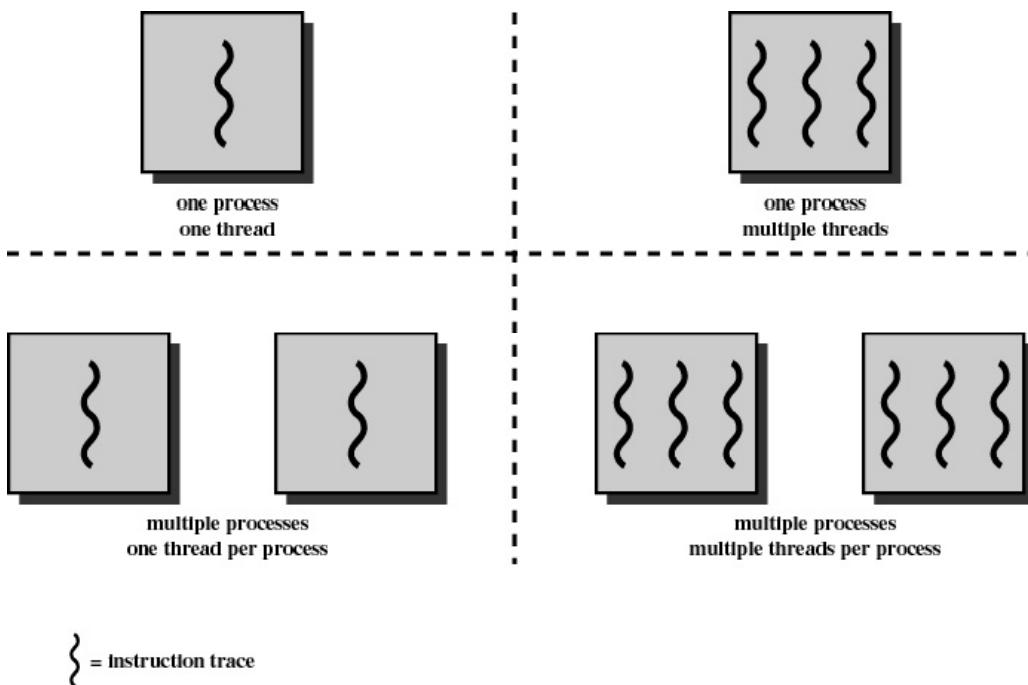


Figure 4.1 Threads and Processes [ANDE97]

Contenido

concepto
estados
PCB
operaciones
threads
ejemplos

Dept. Arquitectura
de Computadores

52 Univ. Málaga

- ➊ **Tipos de threads:**
 - Threads a nivel de usuario
 - Threads a nivel de kernel
- ➋ **Threads a nivel de Usuario**
 - Toda la gestión de los threads se realiza desde la aplicación (a través de biblioteca de usuario que actúa como un 'kernel en miniatura')
 - Ventajas:
 - Alto rendimiento al no consumir recursos kernel (no hacen llamadas al sistema).
 - Comutación entre threads muy rápida: el switch no necesita el paso a modo kernel (evita el 'doble switch').
 - Planificación específica a la aplicación (no depende del SO).
 - Funciona en cualquier SO mientras esté disponible la librería
 - Desventajas:
 - Si un thread se bloquea, bloquea al proceso completo
 - No puede sacar provecho de un multiprocesador

Contenido

concepto
estados
PCB
operaciones
threads
ejemplos

Dept. Arquitectura
de Computadores

53 Univ. Málaga

Threads a nivel de Kernel

- Ejemplos: Windows XP, Linux y OS/2
- El kernel mantiene información de contexto de los procesos y threads
- La planificación se realiza en base a los threads
- Ventajas:
 - un thread en espera de E/S no bloquea al resto de threads del mismo proceso
 - puede aprovecharse de los multiprocesadores
- Desventajas:
 - el switch es más lento que el caso de threads a nivel de usuario
 - la planificación la fija el SO.

Contenido

concepto
estados
PCB
operaciones threads
ejemplos

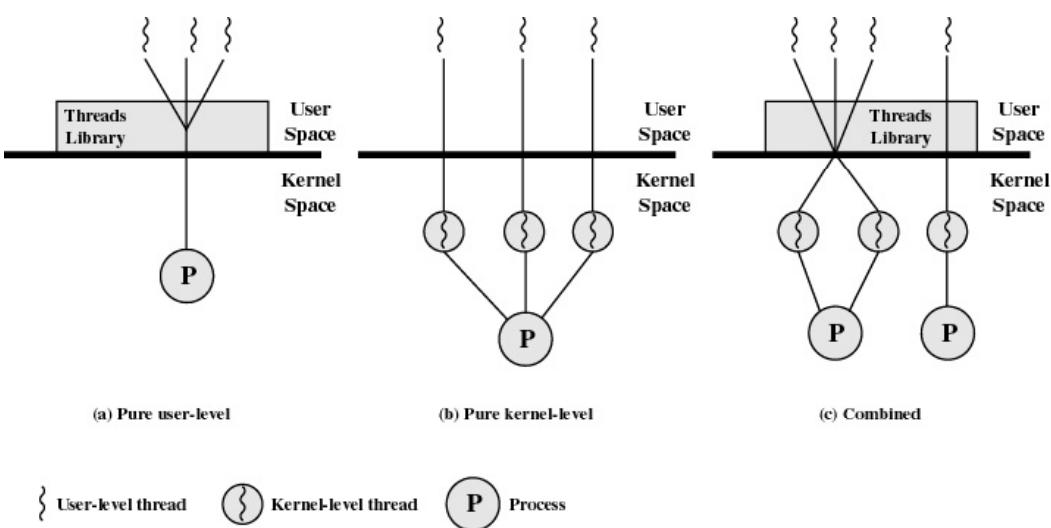


Figure 4.6 User-Level and Kernel-Level Threads

Dept. Arquitectura
de Computadores

54 Univ. Málaga

Contenidoconcepto
estados

PCB

operaciones

threads

ejemplos

Dept. Arquitectura
de Computadores

55 Univ. Málaga

- ➊ Estados del thread: run, ready, block
 - Threads kernel: varios procesadores -> varios en run
- ➋ Estado del proceso: Depende de los threads que contiene

Threads a nivel de usuario

Run: Si tiene algún thread en ejecución y ninguno bloqueado

Ready: Todos los threads están ready (no hay ningún thread en run ni bloqueado)

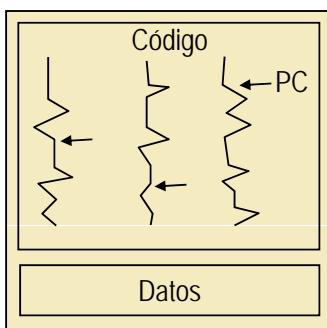
Block: si hay algún thread bloqueado

Threads a nivel de kernel

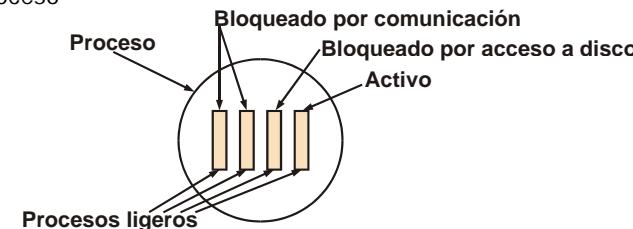
Run: si tiene algún thread en ejecución

Ready: si no hay ningún thread en run y alguno/s ready

Block: si todos los thread están block



Proceso

**Contenido**concepto
estados

PCB

operaciones

threads

ejemplos

Dept. Arquitectura
de Computadores

56 Univ. Málaga

➌ Ventajas del uso de threads

- Comutación de threads más rápida
- Crear y terminar threads es más rápido
- Baja el n° de comutación de procesos
- Simplifica la programación
 - Cuando un algoritmo tiene varias líneas de ejecución
 - Ejemplo: un simulador de vuelo
- Mejora la interactividad
 - Mientras un thread atiende al usuario otro procesa
- Comunicación eficiente entre threads
 - Ya que comparten memoria
- Eficiencia en modelos cliente-servidor
- Soporte SMP (symmetric multiprocessing)

Contenido

concepto

estados

PCB

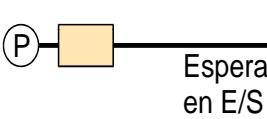
operaciones

threads

ejemplos

Dept. Arquitectura
de Computadores**④ Paralelismo**

Procedimiento 1



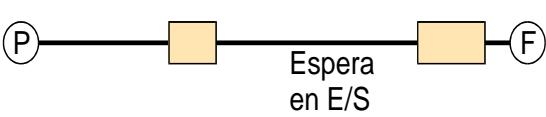
Procedimiento 2

Ejecución serie

Procedimiento 1



Procedimiento 2

**Ejecución paralela****Contenido**

concepto

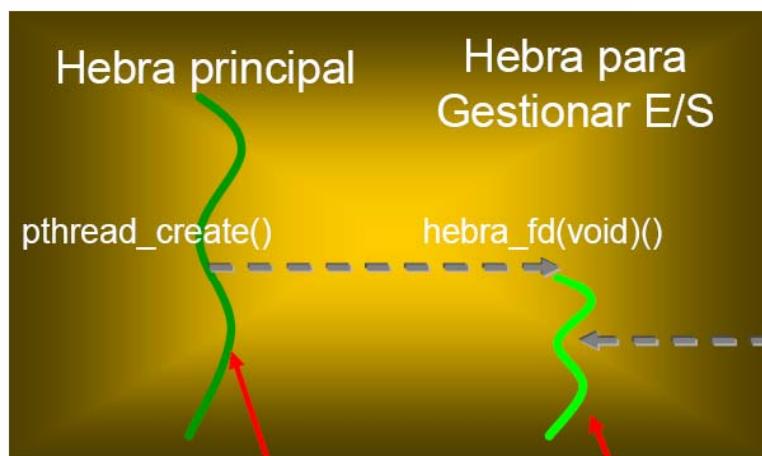
estados

PCB

operaciones

threads

ejemplos

Dept. Arquitectura
de Computadores**⑤ E/S asíncrona****Proceso multihebrado****E/S asíncrona para la hebra principal****E/S síncrona para la hebra que gestiona la E/S**

Contenido

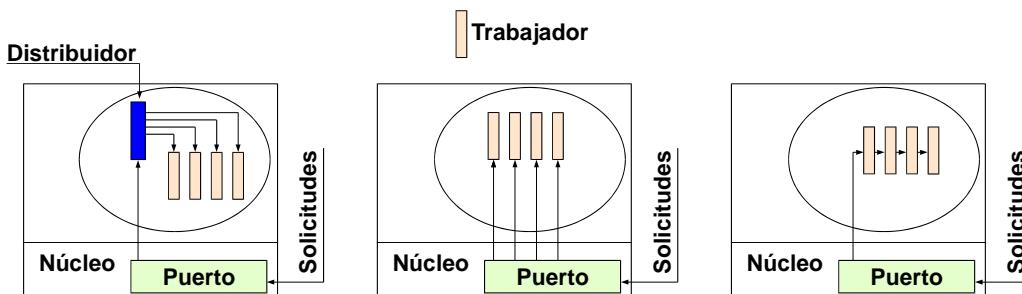
concepto
estados
PCB
operaciones
threads
ejemplos

Dept. Arquitectura
de Computadores

59 Univ. Málaga

E) Diseño de servidores con threads

- Thread distribuidor
 - Cuando llega una petición crea un hijo para hacer la tarea
- Todos los threads iguales
 - Cuando llega una solicitud la atiende un thread libre
- Segmentación
 - Cada thread hace una parte del trabajo (pipeline)



Fuente: Sistemas Operativos. Una visión aplicada.

Contenido

concepto
estados
PCB
operaciones
threads
ejemplos

Dept. Arquitectura
de Computadores

60 Univ. Málaga

E) POSIX (Pthreads)

- ISO/IEEE estándar
- API común
- Casi todos los UNIX tiene una biblioteca de threads

E) Win32

- Muy diferente a POSIX
- Existen bibliotecas comerciales de POSIX

E) Solaris

- Anterior a POSIX

E) Hebras Java

Contenido

concepto
estados
PCB
operaciones
threads
ejemplos

Dept. Arquitectura
de Computadores

61 Univ. Málaga

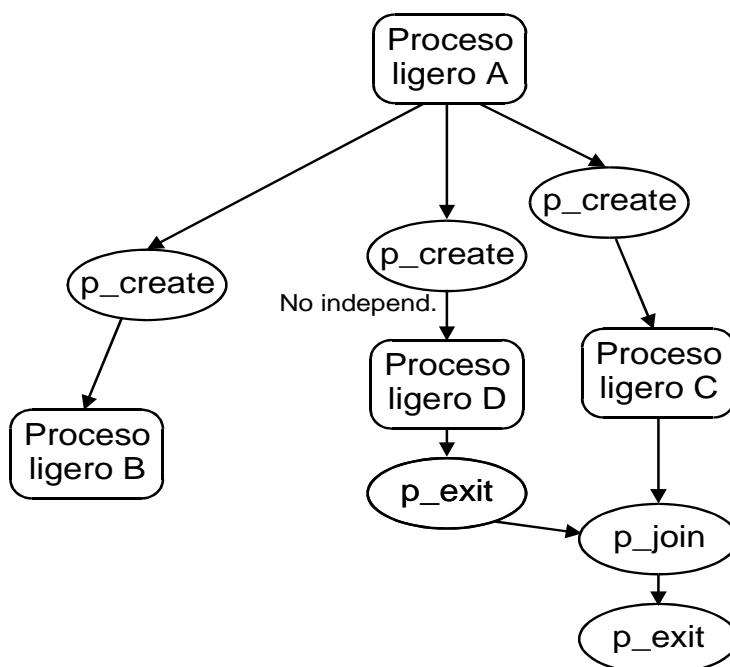
- ④ **int pthread_create (pthread_t *thread, const pthread_attr_t *attr, void *(*func)(void *), void *arg)**
 - Crea un proceso ligero que ejecuta "func" con argumento "arg" y atributos "attr".
 - Los atributos permiten especificar: tamaño de la pila, prioridad, política de planificación, etc.
 - Existen diversas llamadas para modificar los atributos.
- ④ **int pthread_join (pthread_t thid, void **value)**
 - Suspende la ejecución de un proceso ligero hasta que termina el proceso ligero con identificador "thid".
 - Devuelve el estado de terminación del proceso ligero.
- ④ **int pthread_exit (void *value)**
 - Permite a un proceso ligero finalizar su ejecución, indicando el estado de terminación del mismo.
- ④ **pthread_t pthread_self (void)**
 - Devuelve el identificador del thread que ejecuta la llamada.

Contenido

concepto
estados
PCB
operaciones
threads
ejemplos

Dept. Arquitectura
de Computadores

62 Univ. Málaga



Fuente: Sistemas Operativos. Una visión aplicada.

Contenido

concepto
estados
PCB
operaciones
threads
ejemplos

Dept. Arquitectura
de Computadores

63 Univ. Málaga

④ Crear un proceso ligero

- `BOOL CreateThread (`
 `LPSECURITY_ATTRIBUTES lpsa,`
 `DWORD cbStack,`
 `LPTHREAD_START_ROUTINE lpStartAddr;`
 `LPVOID lpvThreadParam,`
 `DWORD fdwCreate,`
 `LPDWORD lpIdThread);`

⑤ Terminar un proceso ligero

- `VOID ExitThread(DWORD dwExitCode);`

Contenido

concepto
estados
PCB
operaciones
threads
ejemplos

Dept. Arquitectura
de Computadores

64 Univ. Málaga

2.6 Ejemplos

Contenido

concepto
estados
PCB
operaciones
threads
ejemplos

Dept. Arquitectura
de Computadores

65 Univ. Málaga

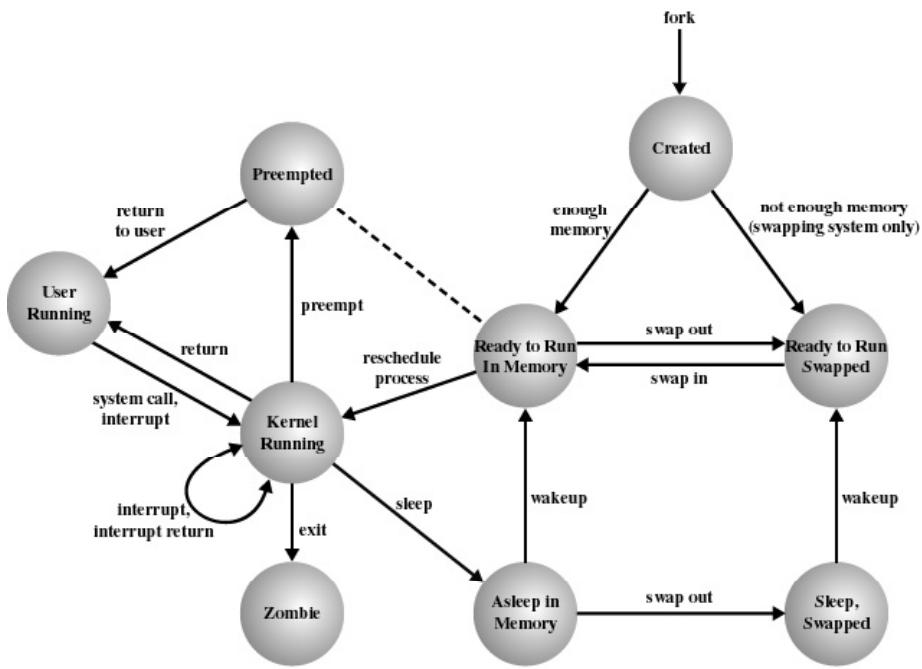


Figure 3.16 UNIX Process State Transition Diagram

Fuente: W. Stallings. Sistemas Operativos

Contenido

concepto
estados
PCB
operaciones
threads
ejemplos

Dept. Arquitectura
de Computadores

66 Univ. Málaga

User Running	Executing in user mode.
Kernel Running	Executing in kernel mode.
Ready to Run, in Memory	Ready to run as soon as the kernel schedules it.
Asleep in Memory	Unable to execute until an event occurs; process is in main memory (a blocked state).
Ready to Run, Swapped	Process is ready to run, but the swapper must swap the process into main memory before the kernel can schedule it to execute.
Sleeping, Swapped	The process is awaiting an event and has been swapped to secondary storage (a blocked state).
Preempted	Process is returning from kernel to user mode, but the kernel preempts it and does a process switch to schedule another process.
Created	Process is newly created and not yet ready to run.
Zombie	Process no longer exists, but it leaves a record for its parent process to collect.

Contenido

concepto
estados
PCB
operaciones
threads
ejemplos

Dept. Arquitectura
de Computadores

67 Univ. Málaga

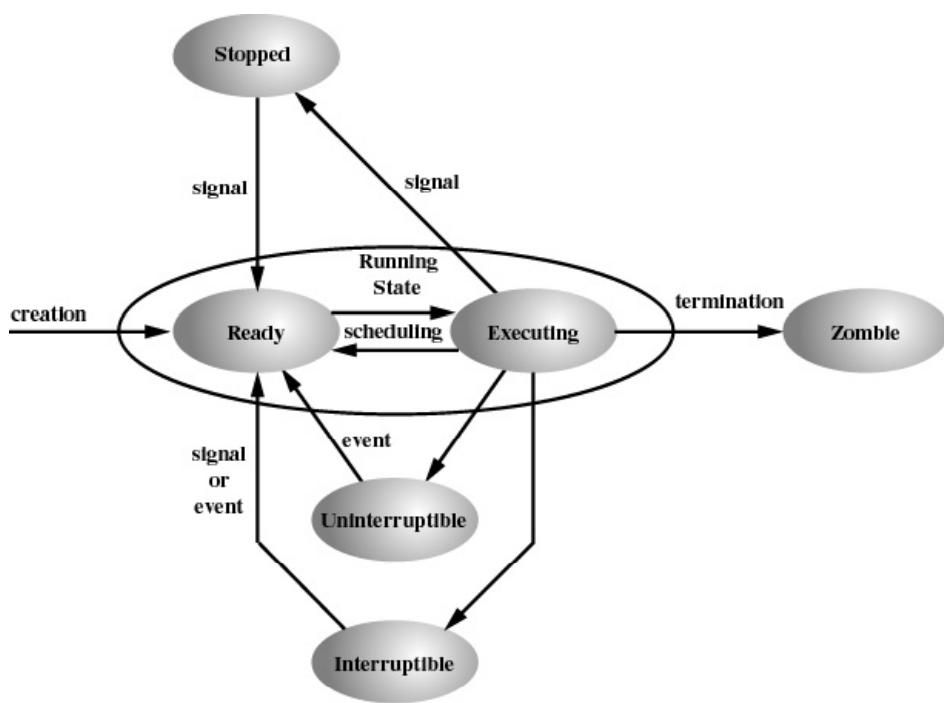


Figure 4.18 Linux Process/Thread Model

Fuente: W. Stallings. Sistemas Operativos

Contenido

concepto
estados
PCB
operaciones
threads
ejemplos

Dept. Arquitectura
de Computadores

68 Univ. Málaga

Estructura de datos task_struct

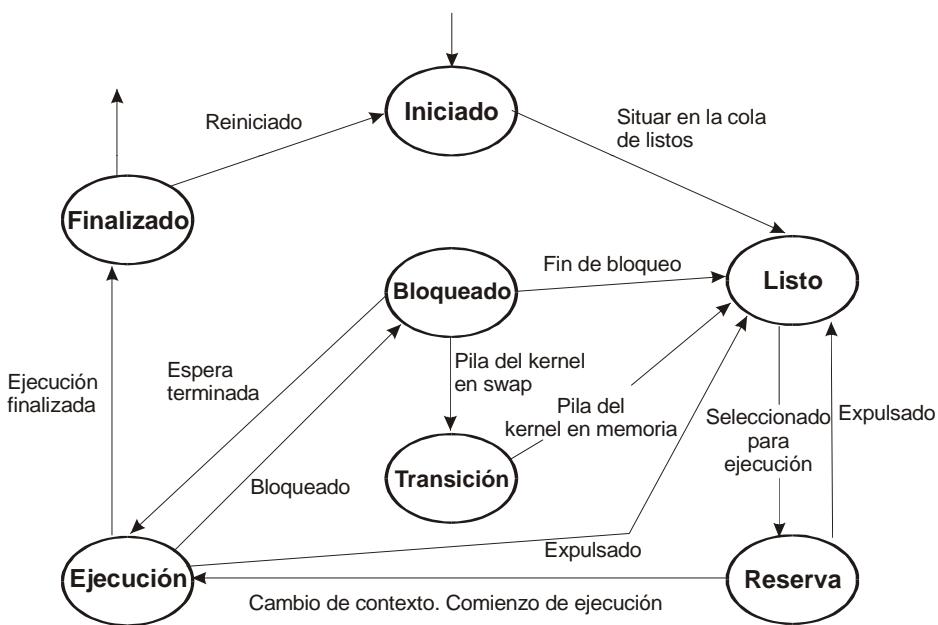
- Estado: Run, Ready, Block, Stoped, Zombie
- Estado del procesador: registros
- Información de planificación
- Identificadores
- IPC: Inter-Process Communication
- Vínculos con padre e hijos
- Tiempos y temporizadores
- Puntero a tablas de archivos
- Puntero a tablas de memoria

Contenido

concepto
estados
PCB
operaciones
threads
ejemplos

Dept. Arquitectura
de Computadores

69 Univ. Málaga



Fuente: Sistemas Operativos. Una visión aplicada.

Contenido

concepto
estados
PCB
operaciones
threads
ejemplos

Dept. Arquitectura
de Computadores

70 Univ. Málaga

Object Type
Object Body Attributes

Object Body Attributes

Services

Process
Process ID
Security Descriptor
Base priority
Default processor affinity
Quota limits
Execution time
I/O counters
VM operation counters
Exception/debugging ports
Exit status

Process
Create process
Open process
Query process information
Set process information
Current process
Terminate process

(a) Process object

Object Type

Object Body Attributes

Services

Thread

Thread ID
Thread context
Dynamic priority
Base priority
Thread processor affinity
Thread execution time
Alert status
Suspension count
Impersonation token
Termination port
Thread exit status

Create thread
Open thread
Query thread information
Set thread information
Current thread
Terminate thread
Get context
Set context
Suspend
Resume
Alert thread
Test thread alert
Register termination port

(b) Thread object

Fuente: W. Stallings. Sistemas Operativos