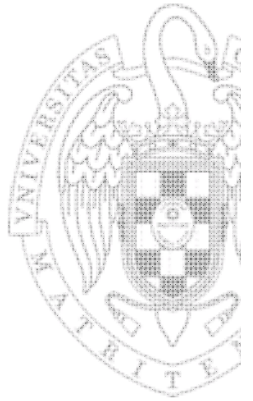
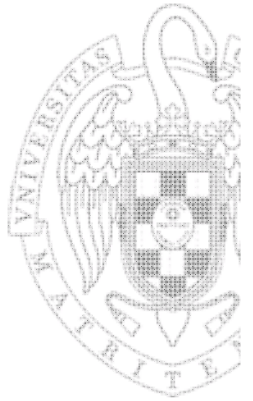


Sistemas Operativos

Procesos e Hilos

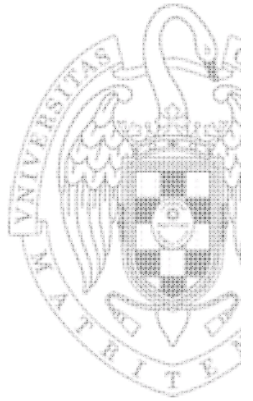
SO

Contenido



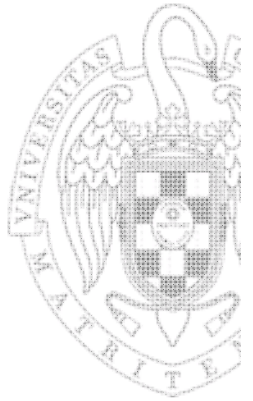
- Procesos
- Multitarea
- Formación de un proceso
- Estados de un proceso
- Información del proceso
- Señales
- Hilos o *threads*

Concepto de proceso



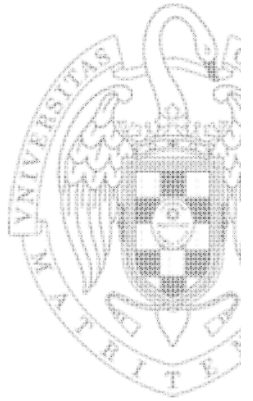
- ¿Qué es un proceso?
  - ¿Qué relación tiene con programa?
  - ¿Qué relación tiene con fichero?

Concepto de proceso



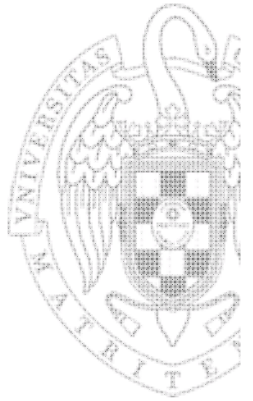
- ¿Un proceso es la unidad mínima de procesamiento?

Concepto de proceso



- Programa: fichero ejecutable en un dispositivo de almacenamiento permanente
- Proceso:
  - Programa en ejecución
  - Unidad de procesamiento gestionada por el SO
    - En realidad, la unidad *mínima* de procesamiento es el hilo (*thread*) ➔ Un proceso puede constar de uno o varios hilos
- Información del proceso:
  - Imagen de memoria: *core image*
  - Estado del procesador: registros del modelo de programación
  - Bloque de control del proceso **BCP**

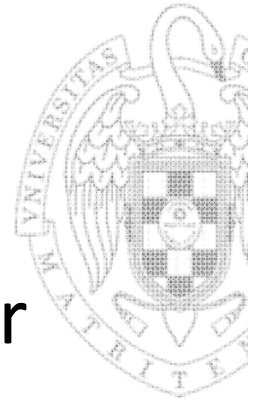
Concepto de proceso



- Mira la carpeta /proc en gnu-linux
  - ¿Qué tiene?
  - ¿Qué significa?

- ¿En Windows?

Curiosidades Linux: /proc

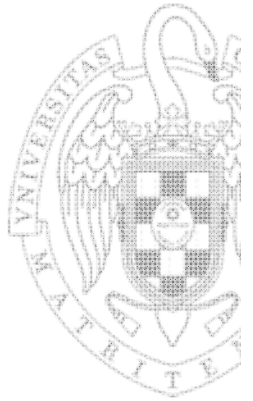


- Directorio /proc contiene un directorio por cada proceso en ejecución
- Permite consultar información sobre el proceso:
  - Línea de comando
  - Mapa de memoria
  - Tabla de páginas
  - …

Proceso *vs.* Ejecutable

- Si tenemos nuestro fichero binario ejecutable *X,* lo ejecutamos y sin esperar a que termine lo volvemos a ejecutar….
  - ¿Tendré uno o dos procesos?
  - Si tengo dos, ¿comparten todas las zonas de memoria?
  - Si uno abre un fichero, ¿el otro ya lo tiene abierto?

Concepto de proceso



- ¿Cuántos procesos se lanzan al iniciar el sistema?
- ¿Por qué?
- ¿Quién los lanza? ¿Cómo?

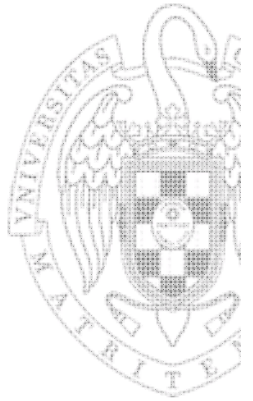
Concepto de proceso

Entender como el Sistema Operativo gestiona los procesos llevará a cualquier programador de aplicaciones o administrador de sistemas a encontrarse en un muy buen escenario para el desempeño de su actividad

Consulta procesos en ejecución

- ps
  - Permite ver la información de todos los procesos en ejecución
  - *man ps* para consultar las múltiples opciones
- top
  - Muestra los procesos en ejecución, refrescando la información periódicamente
  - Permite interaccionar con los procesos (enviar señales)

Contenido

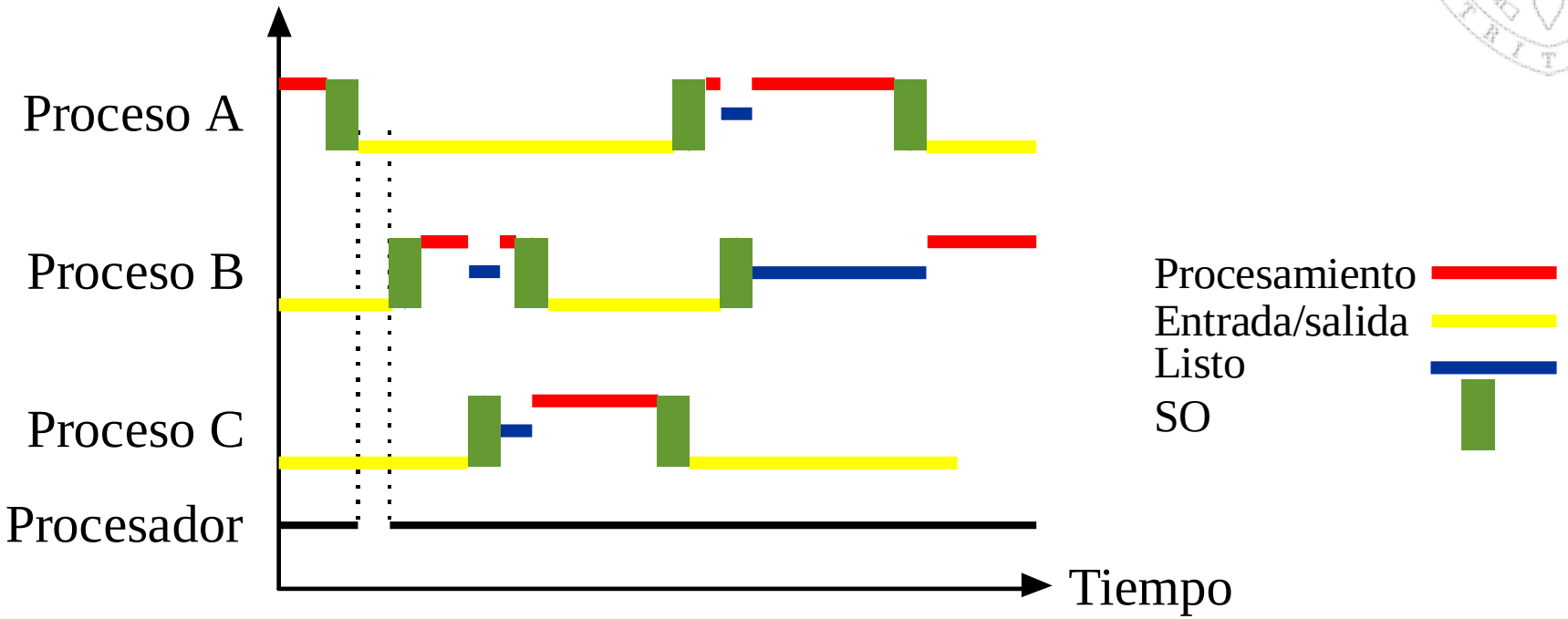


- Procesos
- Multitarea
- Formación de un proceso
- Estados de un proceso
- Información del proceso
- Señales
- Hilos o *threads*

Multitarea

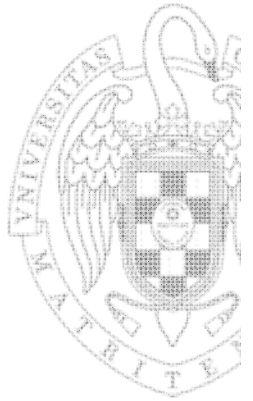
- ¿Qué es?
- ¿Para qué sirve?

Ejecución en un sistema multitarea



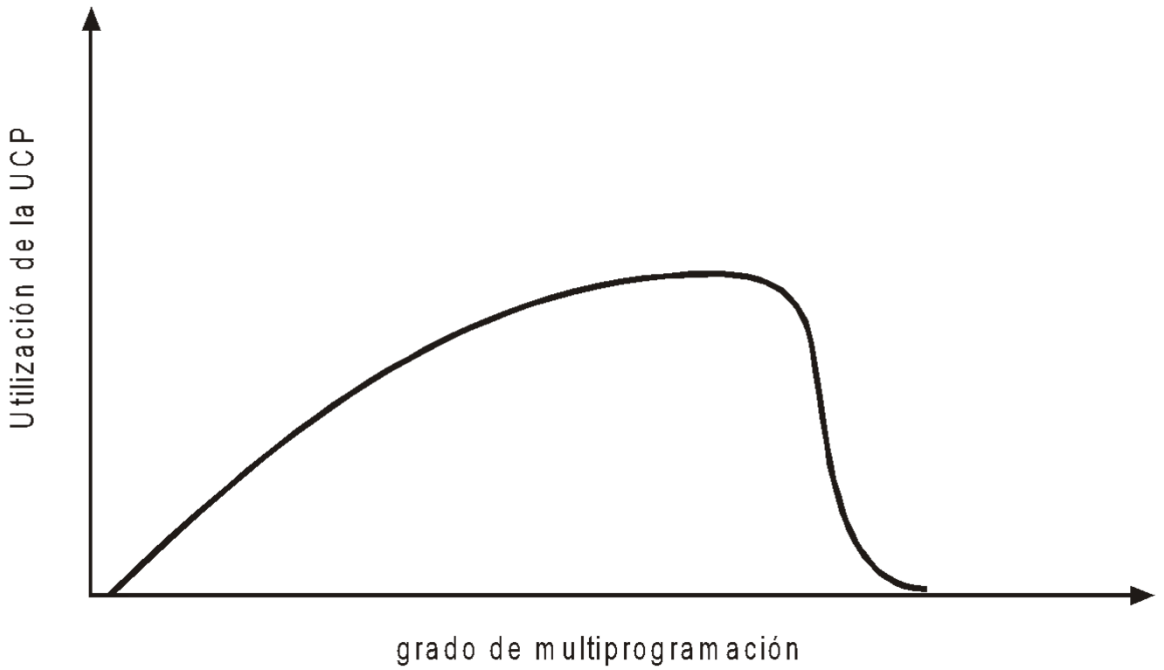
- Proceso nulo o *idle*

Ventajas de la multitarea

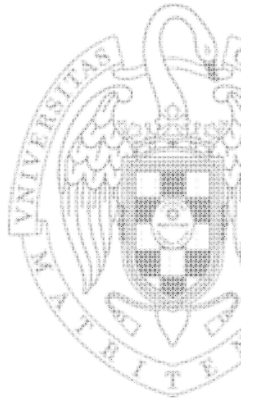


- Facilita la programación, dividiendo los programas en procesos (modularidad)
- Permite el servicio interactivo simultáneo de varios usuarios de forma eficiente
- Aprovecha los tiempos que los procesos pasan esperando a que se completen sus operaciones de E/S
- Aumenta el uso de la CPU

Limites de la multitarea

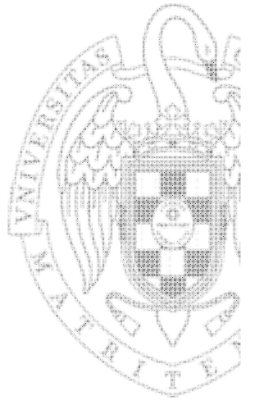


Contenido



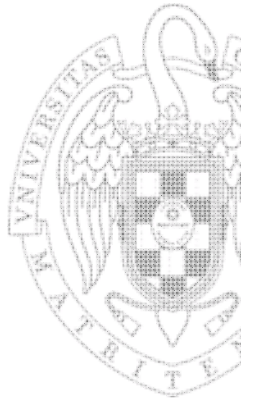
- Procesos
- Multitarea
- Formación de un proceso
- Estados de un proceso
- Información del proceso
- Señales
- Hilos o *threads*

Procesos



- Creación de un proceso
  - Arranque del sistema
  - La ejecución, desde un proceso, de la llamada al sistema para crear un proceso
  - La petición de un usuario para crear un proceso
    - Doble click sobre un programa
  - El inicio de un trabajo por lotes
- Terminación de un proceso
  - Salida normal (voluntaria)
  - Salida por error (voluntaria)
  - Error fatal (involuntaria)
  - Eliminado por otro proceso (involuntaria)
- Estado de un proceso
  - En ejecución
  - Bloqueado
  - Listo

Procesos

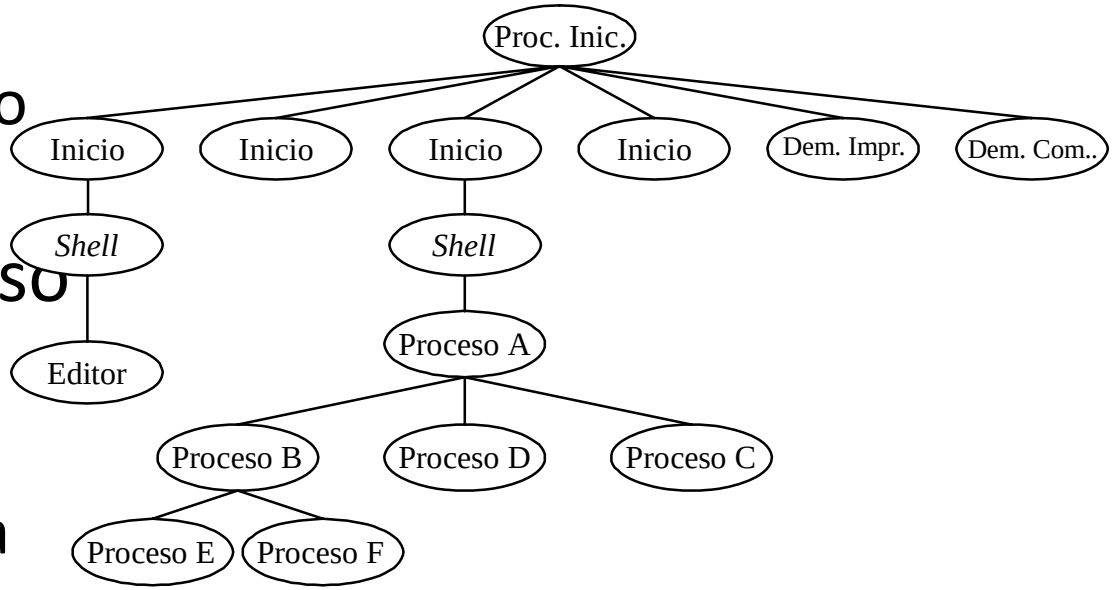


- Como hemos visto, cuando se arranca el sistema operativo se crean varios procesos
  - Procesos en primer plano, interactúan con el usuario
  - Procesos en segundo plano, no están relacionados con usuarios específicos sino con una función específica, **demonios** (daemons)

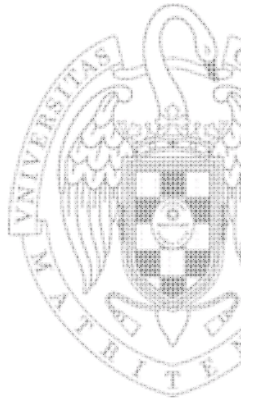
¿Qué hace el demonio cron?

Jerarquía de procesos (UNIX)

- Familia de procesos
  - Proceso hijo
  - Proceso padre
  - Proceso hermano
  - Proceso abuelo
- Vida de un proceso
  - Crea
  - Ejecuta
  - Muere o termina
- Ejecución del proceso
  - Batch
  - Interactivo



Contenido

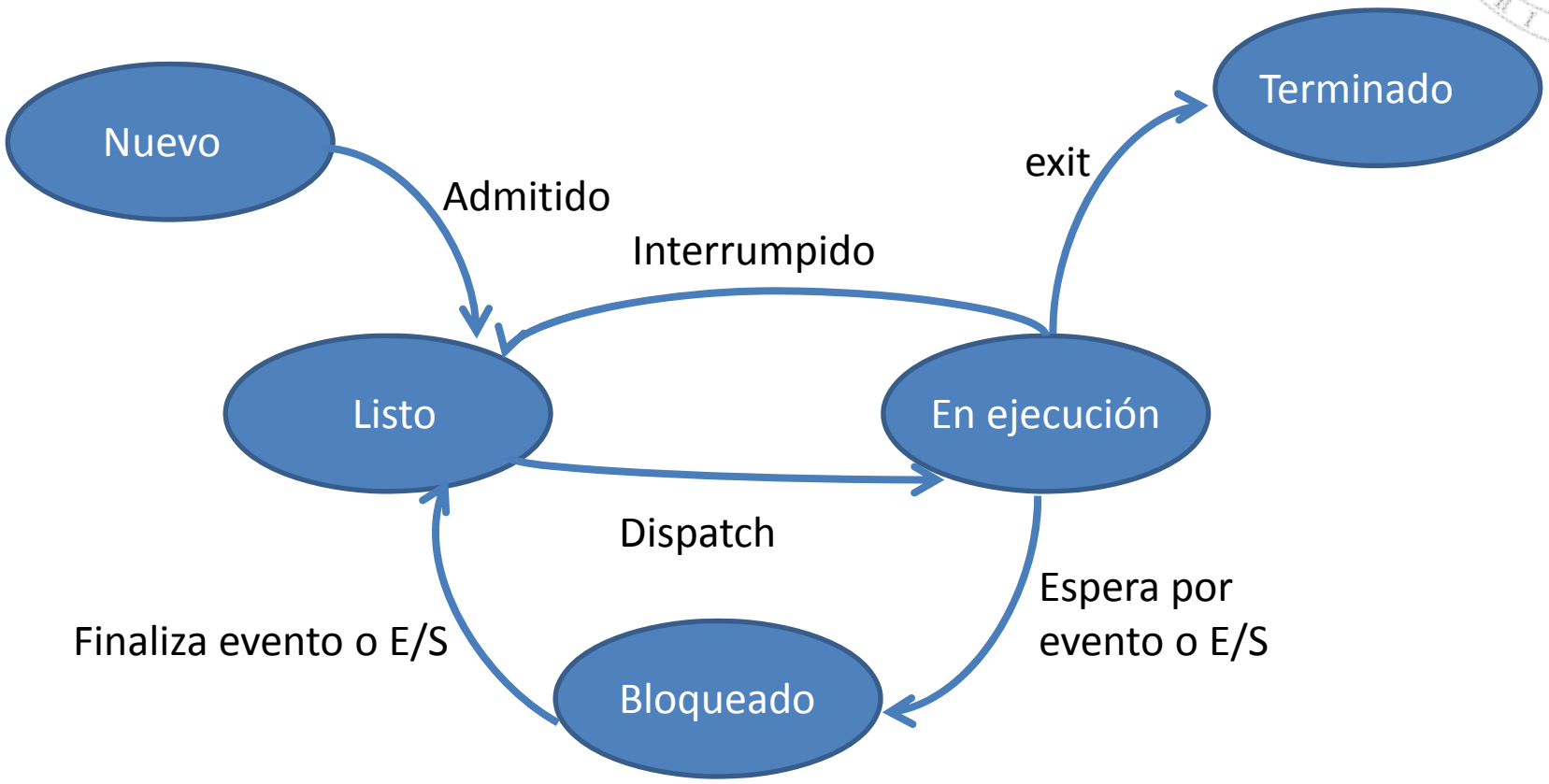


- Procesos
- Multitarea
- Formación de un proceso
- Estados de un proceso
- Información del proceso
- Señales
- Hilos o *threads*

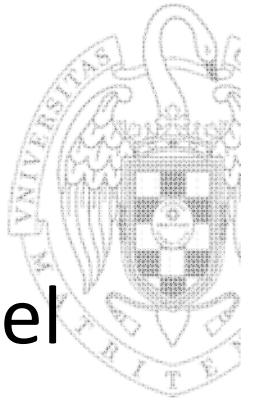
Estados básicos de un proceso

- Planificador: Componente del SO que decide qué proceso se ejecuta en cada procesador y en qué instante
- Proceso nulo o *idle* (uno por cada procesador)

Estados del proceso

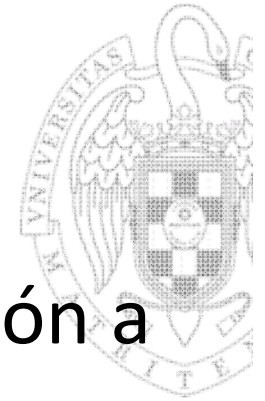


Estados del proceso



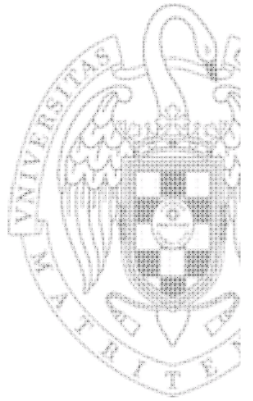
- Para implementar el modelo de procesos, el sistema operativo mantiene una tabla de procesos, con sólo una entrada por cada proceso

Estados del proceso



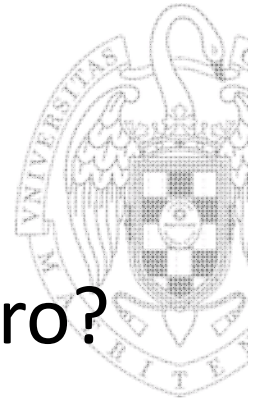
- Qué pasa si un proceso pasa de en Ejecución a Bloqueado
- Cómo somos capaces de volver al punto exacto dónde estaba ese proceso

Contenido



- Procesos
- Multitarea
- Formación de un proceso
- Estados de un proceso
- Información del proceso
- Señales
- Hilos o *threads*

Procesos

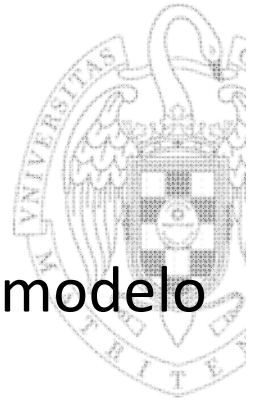


- ¿Cómo se puede pasar de un proceso a otro?
- ¿Cómo no perder información?

Cambio de contexto

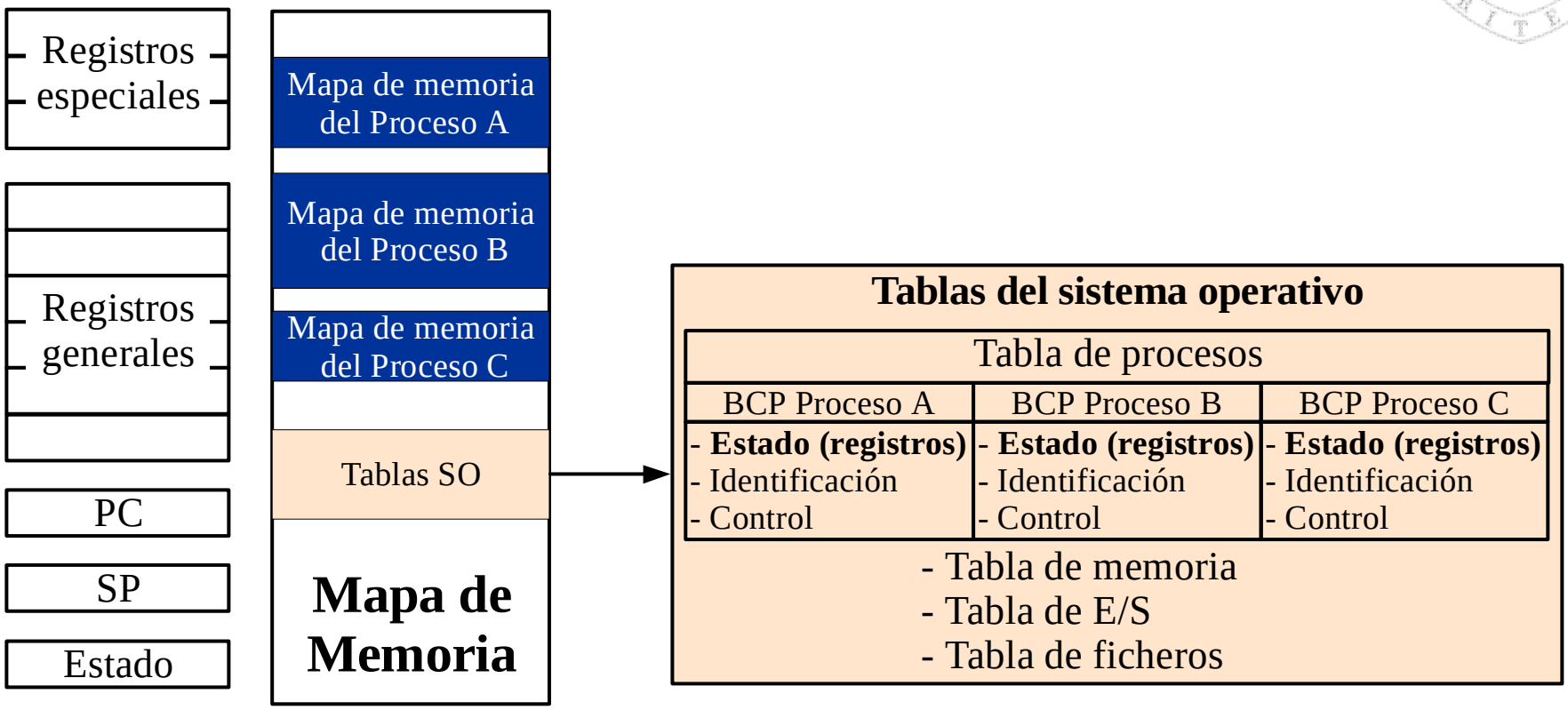
- El **cambio de contexto** es el conjunto de acciones que realiza el SO para cambiar el proceso que está actualmente en ejecución en una CPU
- Acciones (simplificación):
  1. Salvar el contexto del proceso saliente (registros del modelo de programación) en su **BCP**
  2. Cambiar el estado del proceso saliente (En ejecución -> Otro Estado)
  3. Intercambio de los espacios de direcciones
     - Segmentos o regiones de memoria que puede usar un proceso
     - En x86. GDT (Global descriptor Table)
     - En algunas arquitecturas → Invalidación de entradas de la TLB
  4. Cambiar el estado del proceso entrante, (Listo -> En ejecución) Restaurar su contexto (BCP -> registros) y volver a modo usuario
- Puede llegar a ser una operación bastante costosa
- ~~*El cambio de modo de ejecución del procesador no siempre desencadena un cambio de contexto*~~

Información de un proceso (I)

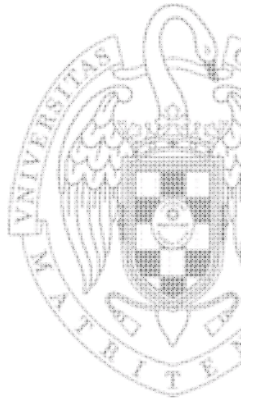


- **Estado del procesador:** contenido de los registros del modelo de programación
- **Imagen de memoria:** contenido de los segmentos de memoria en los que reside el código y los datos del proceso
- **Bloque de control del proceso (BCP) o Descriptor de proceso**
  - Estado actual del proceso
  - Estado del procesador
    - Actualizado cuando proceso no se está ejecutando en la CPU
  - Identificadores *pid*, *uid*, etc.
  - Prioridad
  - Segmentos de memoria (espacio de direcciones)
  - Ficheros abiertos
  - Temporizadores
  - Señales
  - Semáforos
  - Puertos

Información de un proceso (II)

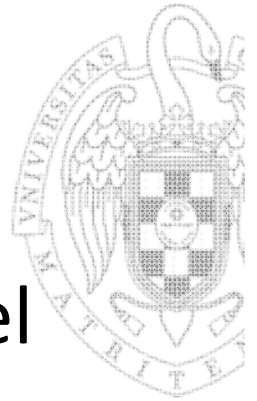


Estado del procesador



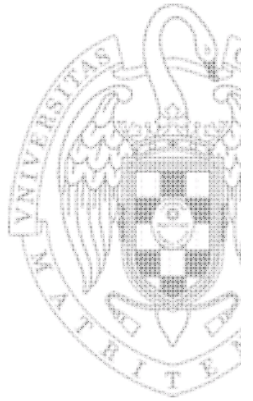
- Está formado por el contenido de todos sus registros:
  - Registros generales
  - Contador de programa
  - Puntero de pila
  - Registro de estado
  - Registros especiales
- Cuando un proceso está ejecutando su estado reside en los registros del computador.
- Cuando un proceso no ejecuta su estado reside en el BCP.

Imagen de memoria



- La imagen de memoria está formada por el conjunto de regiones de memoria que un proceso está autorizado a utilizar
- Si un proceso genera una dirección que esta fuera del espacio de direcciones el HW genera una excepción que el SO captura
- La imagen de memoria, dependiendo del computador, puede estar referida a memoria virtual o memoria física

Información del BCP

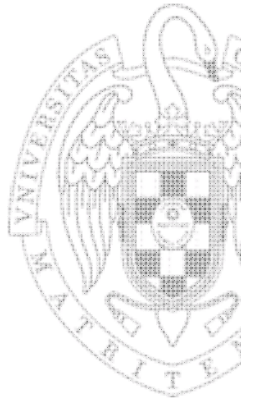


- Información de identificación:
  - PID del proceso, PID del padre (PPID)
  - ID de usuario y grupo reales (uid/gid reales)
  - ID de usuario y grupo efectivos (uid/gid efectivos)
- Estado del procesador
- Información de control del proceso:
  - Información de planificación y estado
  - Descripción de los segmentos de memoria del proceso
  - Recursos asignados (ficheros abiertos, ...)
  - Recursos de comunicación entre procesos
  - Punteros para estructurar los procesos en listas o colas

Información del BCP II

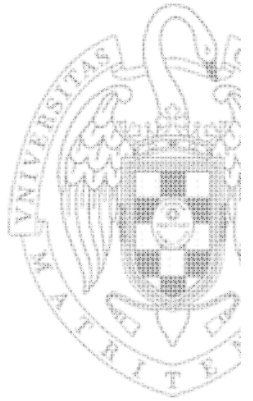
- Información fuera del BCP
  - Conveniente por implementación (la consideramos del BCP)
  - Para compartirla
- La tabla de páginas se pone fuera
  - Describe la imagen de memoria del proceso
  - Tamaño variable
  - El BCP contiene el puntero a la tabla de páginas
  - La compartición de memoria requiere que sea externa al BCP
- Punteros de posición de los ficheros
  - Si se añaden a la tabla de ficheros abiertos (en el BCP) no se pueden compartir
  - Si se asocian al nodo-i se comparte siempre
  - Se ponen en una estructura común a los procesos y se asigna uno nuevo en cada servicio OPEN

Tablas del sistema operativo



- **Tabla de procesos** (tabla de BCP)
- **Tabla de memoria**: información sobre el uso de la memoria.
- **Tabla de E/S**: guarda información asociada a los periféricos y a las operaciones de E/S
- **Tablas de fichero**: información sobre los ficheros abiertos.

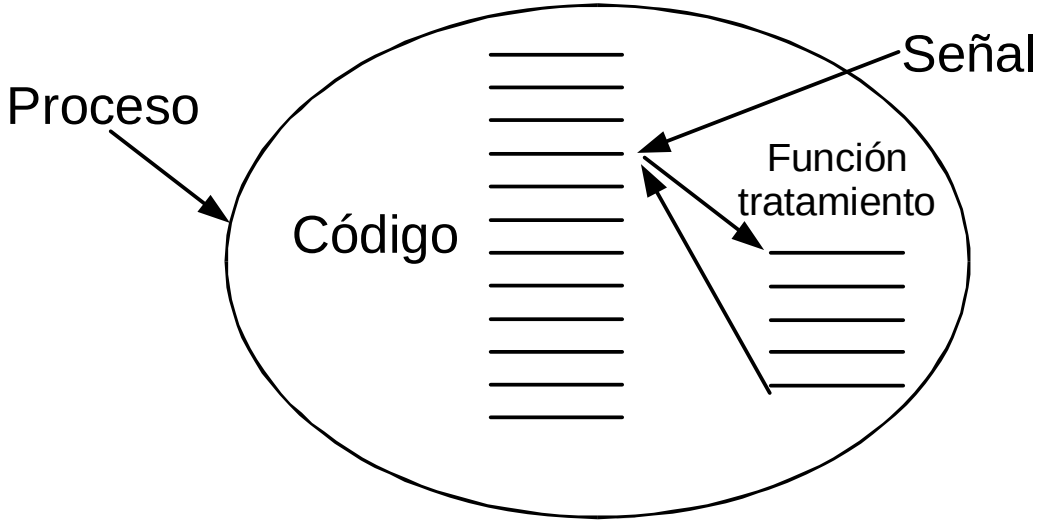
Contenido



- Procesos
- Multitarea
- Formación y estados de un proceso
- Información del proceso
- Señales
- Hilos o *threads*

Señales

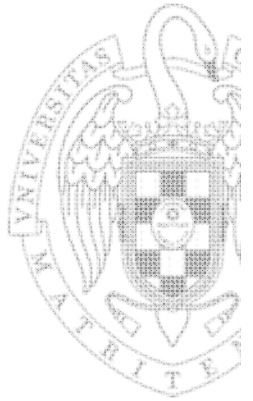
- Las señales son interrupciones al proceso
- Envío o generación:
  - Proceso → Proceso (con mismo *uid*) con *kill*
  - SO → Proceso



Señales II

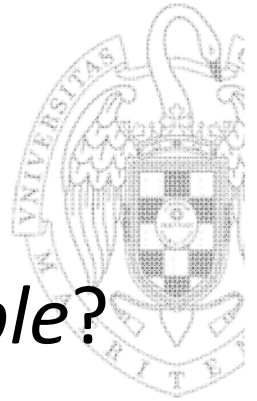
- Hay muchos tipos de señales, según su origen
  - SIGILL instrucción ilegal
  - SIGALRM vence el temporizador
  - SIGKILL mata al proceso
- El SO las transmite al proceso
  - El proceso debe estar preparado para recibirla
    - Especificando un *manejador* de señal con *sigaction*
    - Enmascarando la señal con *sigprogmask*
  - Si no está preparado → acción por defecto
    - El proceso, en general, muere
    - Hay algunas señales que se ignoran o tienen otro efecto
- El servicio *pause* para el proceso hasta que recibe una señal

Contenido



- Procesos
- Multitarea
- Información del proceso
- Formación y estados de un proceso
- Señales
- Hilos o *threads*

Preguntas Hilos

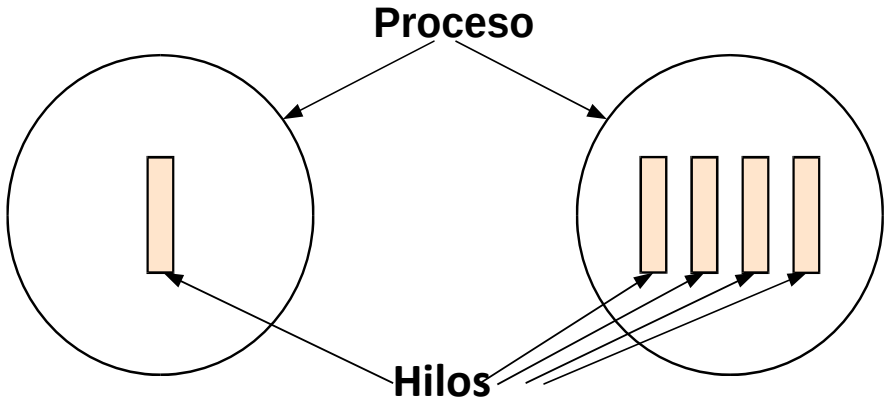


- ¿Un proceso es la *unidad mínima ejecutable*?
- ¿Por qué no?

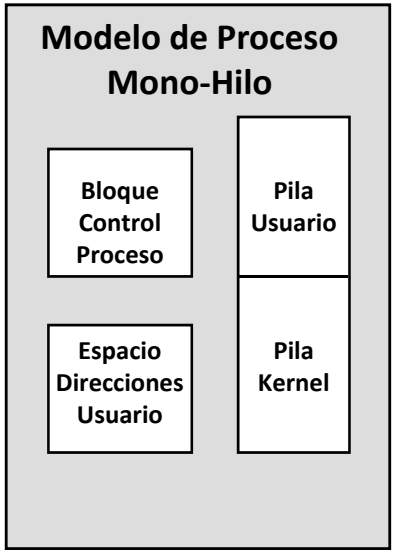
Qué es un hilo

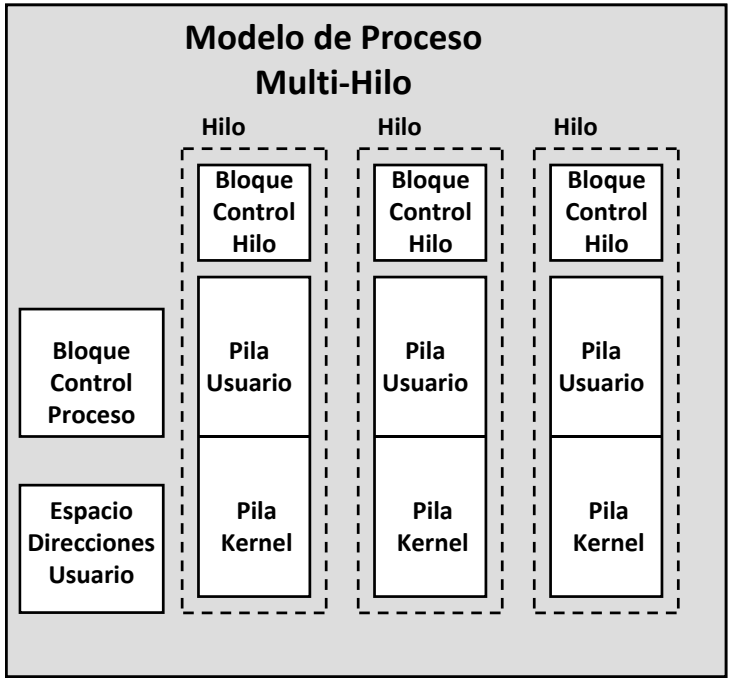
Hilos o threads

- Por Hilo
  - Contador de programa, Registros
  - Pila
  - Estado (ejecutando, listo o bloqueado)
  - Bloque de control de *thread*
- Por proceso
  - Espacio de direcciones de memoria
  - Variables globales
  - Ficheros abiertos
  - Procesos hijos
  - Temporizadores
  - Señales y semáforos

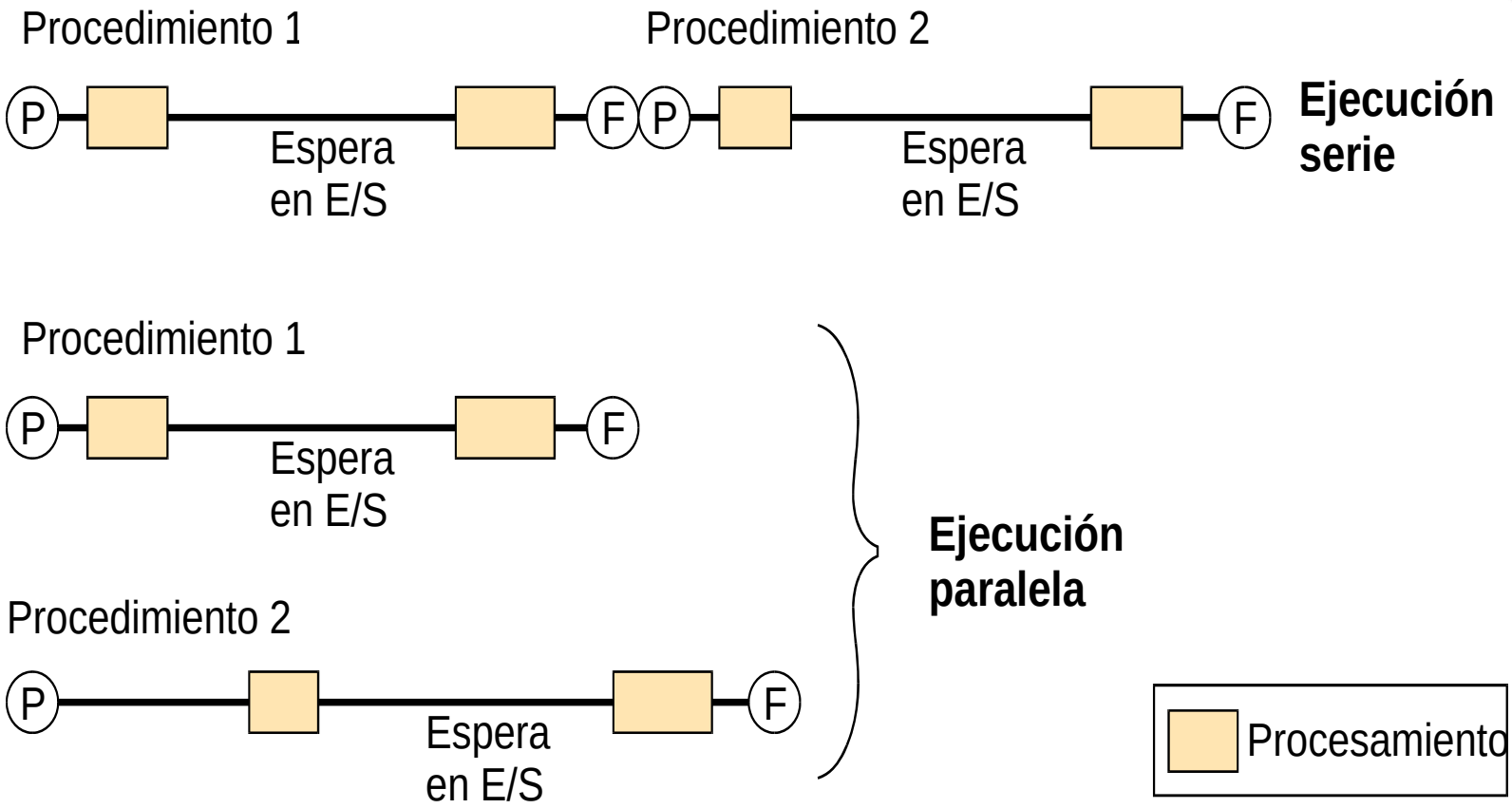


Mono-hilo vs Multi-hilo

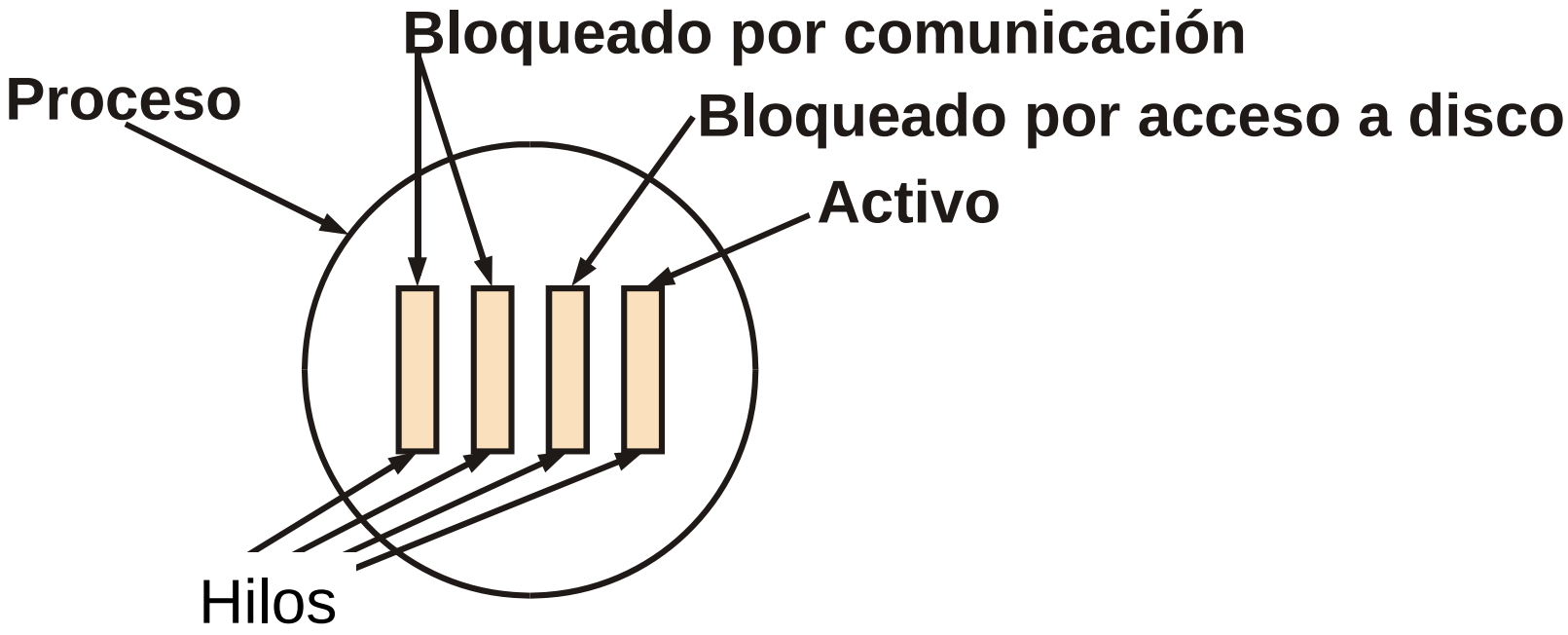




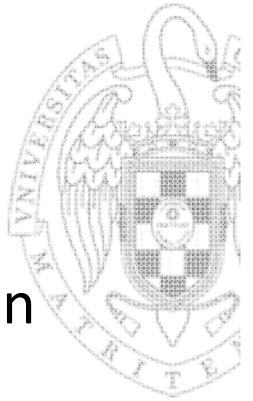
Paralelización utilizando hilos



Estados de un hilo

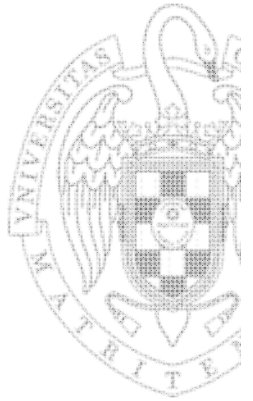


Ventajas threads vs. procesos



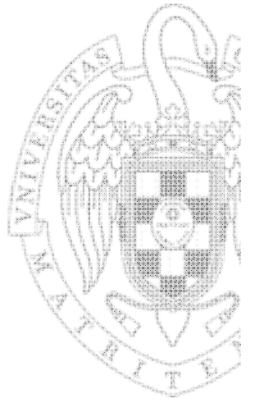
- Tiempo de procesador para operaciones relacionadas con creación, destrucción, planificación y sicronización:
  - 10 hils *vs* 100 proceso.
- El cambio de contexto entre hilos (de kernel) de un mismo proceso es menos costoso ➔ No es necesario cambiar el espacio de direcciones “activo” de usuario
- Permiten compartir memoria entre ellos de forma fácil y eficiente
  - ¡¡Todos tienen el mismo espacio de direcciones!!

Diseño con hilos



- Permite separación de tareas
- Paralelismo
  - Aumenta la velocidad de ejecución del trabajo
- Programación concurrente (memoria compartida)
  - Variables o estructuras de datos compartidas
  - Funciones reentrantes
    - Imaginar otra llamada al mismo código
  - Mecanismos de sincronización entre hilos (mutex, semáforos,…)
  - Variables globales
  - Simplicidad vs exclusión en el acceso

Alternativas al diseño multihilo



- Proceso con un solo hilo
  - No hay paralelismo
    - Llamadas al sistema bloqueantes
  - Paralelismo gestionado por el programador
    - Llamadas al sistema no bloqueantes
- Múltiples procesos convencionales cooperando
  - Permite paralelismo
  - No comparten variables
  - Mayor sobrecarga de ejecución