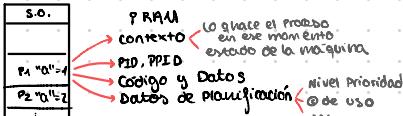
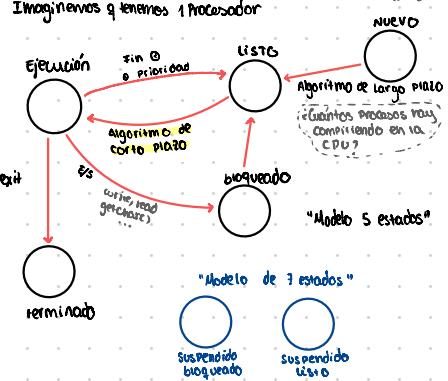


T4: Procesos e hilos

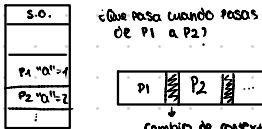
INTRODUCCIÓN



Imaginemos q tenemos 1 procesador → solo 1 proceso cada vez

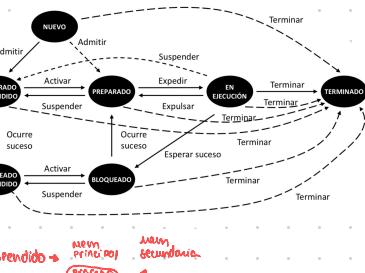


PS entre Proceso e Hilo



Hilos → comparten memoria → no requiere cambios de contexto

Desventaja: Puede haber división de 2 hilos q traten una = variable



IMPLEMENTACIÓN DE PROCESOS

• Requiere

- Bloque de control del proceso → PCB [almacena info de 1 proceso]
- Bloque de control del sistema → SCS [almacena info de todos los procesos]
- Asignar los recursos necesarios → ejecución
- Multiplexar la CPU entre los procesos
 - ↳ simular la ejecución // concurrencia virtual

• PCB

- ↳ info q almacena = 3 categorías
 - identif. proceso de proceso: PID, PPID, identif. usuario
 - Contexto (info. estadio proceso): conteniente de registros del procesador "PSW" Program Status Word
 - Info. del control del proceso: memoria para q s.o. controle y coordine q.s. procesos
- Identificador del proceso
Información de estado del proceso
Información de control del proceso
- Plano de memoria
- Espacio privado del controlador de memoria (programa + datos)
- Espacio compartido de direcciones
- Identificador del proceso
Información de estado del proceso
Información de control del proceso
- Plano de memoria
- Espacio privado del controlador de memoria (programa + datos)
- Espacio compartido de direcciones
- Registros de CP, CR, códigos condición
Info. resultado, numero fila (LIFO)
- Estado proceso (esperando/bloqueado), prioridad, info. planificación, suceso (identidad proceso generada para reanudarse), hora de alta

Creación de un proceso

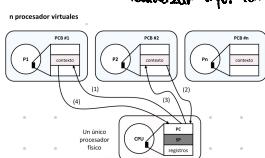
1. Asignar un identificador
2. Retener estadio para el proceso
3. inicializar PCB
4. Establecer entares apropiados
 - ej: proceso → cosa de listas
5. Creación / expansión otras estruct. de datos

Implementación varios procesos

Ejecutan aparentemente simultáneamente → > 0 CPU

• Requiere cambios de contexto

- ↳ Salvar contexto
- ↳ TCU → nuevo contexto
- ↳ Actualizar info. control procesos



T4: Procesos e hilos

Criterios de Planificación

- Comunes
 - Equidad
 - Prioridad
 - Balancio → todos recursos del SIST deben estar garantizados
- Sist. Por lotes
 - Producción
 - @ de retorno
 - Utilización CPU
- Sist. Interactivos
 - Q de respuesta
 - (más prioridades) → Prioritariidad: procesos simples deben responder rápidamente
- Sist. De Q real
 - cumplir plazos
 - Previsibilidad

Medidas de efectividad

- para ver si se cumplen los criterios
- Utilización → @ recursos ocupados
 - Productividad → @ trabajos terminados
 - @ de retorno → @ salida - @ entrada = Total - Tiem. Total / Total
 - Q. Espera → Q. Proceso actual = P1
 - Q. Respuesta → Q. desde momento solicitud → ejecuta
 - Equidad → Proporción justa CPU (prioridad + "interactivos")

Optimización criterios de planificación

- Sist. Por lotes
 - Maximizar utilidad
 - Prioridad
 - minimizar: @ de espera
- Sist. Interactivos
 - Equidad
 - Q de respuesta predecible
- Efecto de la Multipropagación

sin multiprogramación



Con multiprogramación: intercalando ráfagas de CPU con ráfagas de I/O



Algoritmos de Planificación

Planificación sin expulsión:

- Una vez CPU asignada → 1 proceso
- La máquina vuelve a bloquear
- Optimiza → Proceso por lotes

Algoritmo FCFS → First Come, First Served (FCFS)

Proceso			Hogado	Tow
P1	0	24		
P2	0	3		
P3	0	3		

Caso 1: Orden de llegada P1, P2, P3			Caso 2: Orden de llegada P2, P1, P3	
T de espera media:	0	17	1	17
T de respuesta media:	24	24	3	24
T de ejecución media:	12	12	12	12

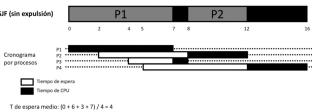
- SIN EXPULSIÓN
- FÁCIL DE IMPLEMENTAR

- DESVENTAJAS
 - No optimiza → @ espera
 - Trabajos largos retrasan cortos
 - No adecuado → Sist. Interact.

Algoritmo SJF → Shortest Job First

- SIN EXPULSIÓN

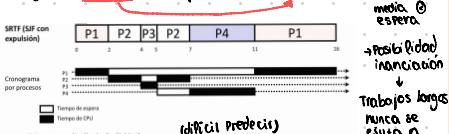
SJF (sin expulsión)			
P1	0	7	
P2	2	4	
P3	4	1	
P4	5	4	



Algoritmo SRTF → Shortest remaining time left

SJF pero con expulsión

Si toca un proceso TEPD < TCPD Actual → PCPU



- OPTIMIZA: media → estancia
- POSTI POCO: inactivación
- TRABAJOS LARGOS: Nunca se ejecutan

Algoritmo RR → Round-Robin (turno alternativo)

- A cada proceso α milis → pequeña cantidad de CPU → "quantum"

• Si: Trabajo > quantum → se expulsa

• Valores quantum de tiempo

↳ 9 grandes (9=20): algoritmo **Round-Robin**, FCFS

↳ 4 pequeños: grande **contexto**, @ necesaria para cambios contexto

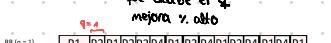
↳ Regla práctica: 80% intervalos CPU + quantum

Propiedades

- n procesos → @ espera más (n-1) q
- @ retorno → varía según q

En general peor q. SJRTF

- Si trabajos acaban antes de que acabe el q. mejora % alto



Plaificación Algoritmos SJF/SRTF

estimación intervalo CPU:

$$S_{n+1} = \alpha T_n + (1-\alpha) S_n$$

(q) expresión

lat. real
lat. estimado
del n-ésimo intervalo

→ CASO $\alpha = 1$ → datos históricos son irrelevantes

→ solo uno trabajo más reciente

→ CASO $\alpha = 0$ → datos históricos relevantes

→ actividad actual es tan transferencia

→ habitualmente $\alpha = 1/2$ → más reciente y antigua se ponderan igualmente

→ $S_{n+1} = \alpha T_n + (1-\alpha) S_n$

$$S_{n+1} = \alpha T_n + (1-\alpha) S_n$$

$$S_{n+1} = \alpha T_n + (1$$

