Comunicación entre Procesos. IPC

Sistemas Operativos

2° año Ing. en Sistemas de Información

Universidad Tecnológica Nacional Facultad Regional Villa María

Comunicación entre Procesos

- Ejemplo conocido?

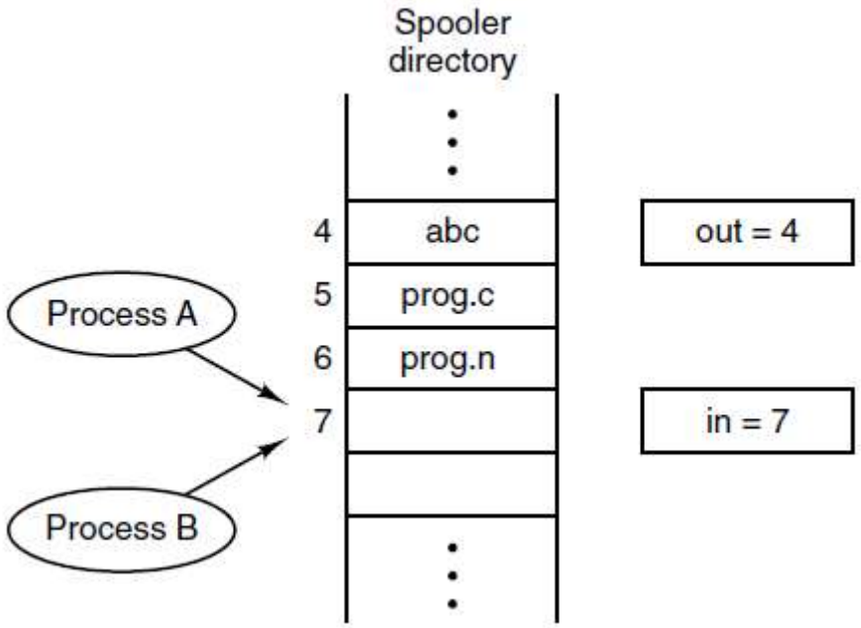
- 3 Cuestiones a resolver:

  – Paso de información
  – Interposición (reserva vuelo)
  – Dependencias

- Qué sucede con los hilos?

Condiciones de carrera

- Dos procesos o mas leen o escriben datos compartidos, el resultado depende de quién y cuando se realiza el cambio.



Regiones críticas

- Exclusión mutua y Regíon crítica

- 4 Condiciones

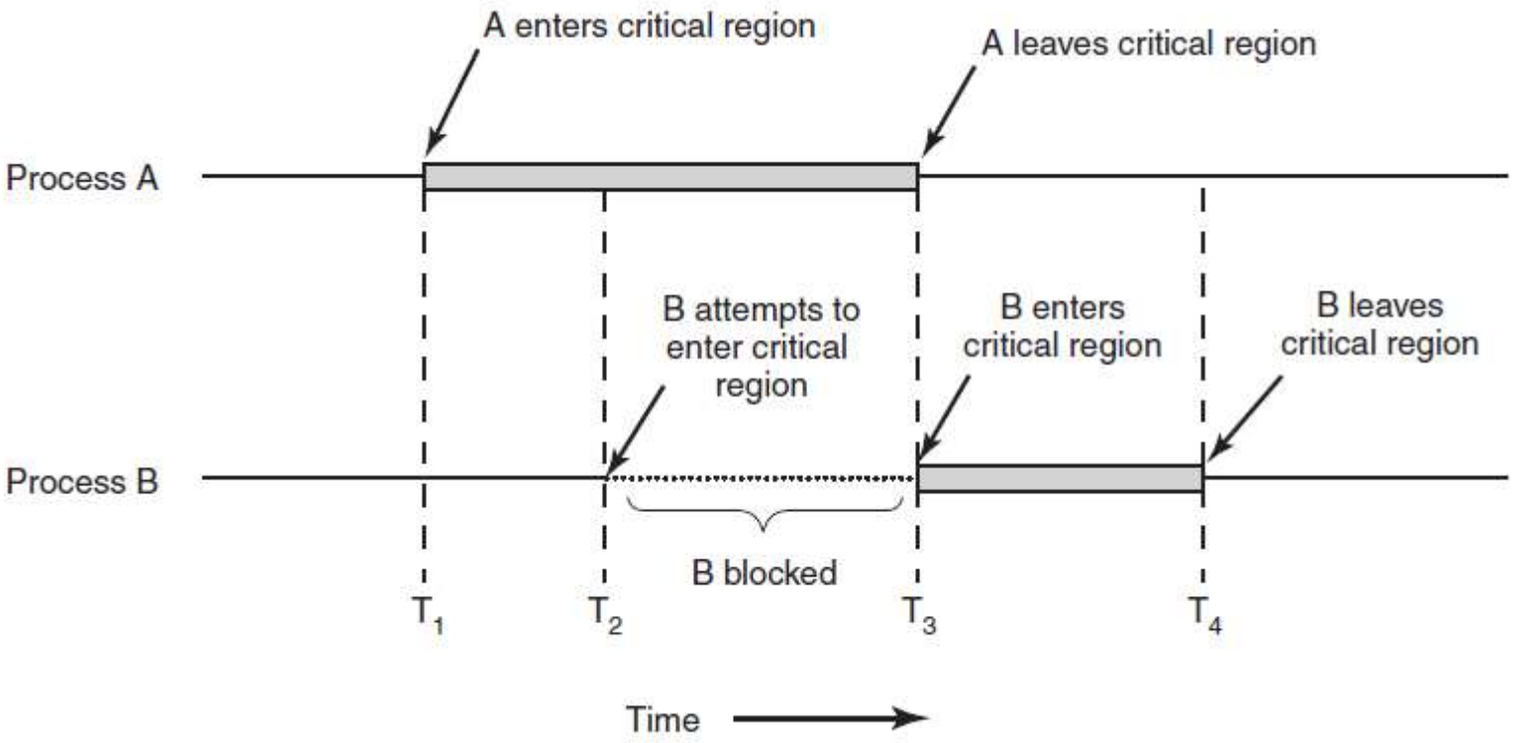
1) No puede haber dos procesos de manera simultánea en sus regiones críticas

2) No pueden hacerse supociones acerca de Velocidad y #cpu

3) Ningún proceso que no este en su sección crítica puede bloquear a otro

4) Ningún proceso tiene que esperar por siempre para entrar en su región crítica

Regiones críticas



Exclusión mutua con mediante el uso de regiones críticas

Exclusión mutua con espera ocupada

Exclusión mutua con espera ocupada

1) Deshabilitar interrupciones

- 1 solo procesador
- Si se cae el proceso?
- Si hay mas núcleos o mas procesadores?

2) Variables candado

- Una variable compartida
- 0 libre
- 1 ocupado
- Mismo problema que el spooler!!!

Exclusión mutua con espera ocupada

Exclusión mutua con espera ocupada

3) Alternancia Estricta

- Qué sucede si un proceso es mucho mas lento que otro?
- Se viola la condición 3. „Ningún proceso que no este en su sección crítica puede bloquear a otro“. Ej, cola de impresión?

```
while (TRUE) {
    while (turn != 0)        /* loop */ ;
    critical_region( );
    turn = 1;
    noncritical_region( );
}
```

(a)

```
while (TRUE) {
    while (turn != 1)        /* loop */ ;
    critical_region( );
    turn = 0;
    noncritical_region( );
}
```

(b)

Exclusión mutua con espera ocupada

Exclusión mutua con espera ocupada

4) Solución de Peterson

- Mejora de la alternancia estricta
- Se utilizan llamadas a procedimientos
- No se viola la condición 3
- Se utilizan variables señal y turno
- Ej Algoritmo de Perteson

Dormir y despertar

Dormir y despertar:

–problema de inversión de prioridades. Sleep bloquea, wake up activa.

El problema del productor – consumidor

–A inserta → Buffer → B extrae

–A duerme cuando buffer lleno. Cuenta = N

–B duerme cuando buffer vacío. Cuenta = 0

–Ambos comprueban si el otro se tiene que despertar, wakeup()

–!!! Si B se quiere dormir pero antes sale, A inserta, no lo puede despertar porque está despierto, luego b duerme y se llena el buffer. Mismo ej. Que el spooler.

–Solución: bit de espera despertar para que B no se vaya a dormir.

Semáforos

- Semáforos:
  - Variable P down o wait→ sleep
  - Variable V up o signal → wakeup
  - Operaciones atómicas
  - Ej: uso de semáforo por dispositivo E/S
  - Ej: Semáforos

Mutexes, Monitores y Pasaje de mensajes

- Mutexes:
  - No cuentan como los semáforos
  - El proceso llama a mutex_lock , mutex_unlock

Mutex actúa como un candado

- Monitores
  - Son construcciones del lenguaje.
  - Convertir todas las regiones críticas en procedimientos de monitor, nunca habrá dos procesos que ejecuten sus regiones críticas al mismo tiempo.

- Pasaje de mensajes

Barreras

- Barreras
  - Ningún proceso puede continuar a la siguiente fase sino hasta que todos los procesos estén listos para hacerlo.