

Laborator 5

Comunicare între Procese

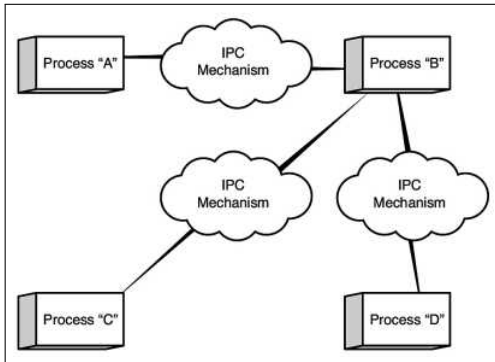
Sisteme de Operare

17 - 23 Martie 2011

- ▶ Procesele din cadrul unui sistem
 - ▶ pot fi independente
 - ▶ pot coopera / colabora

- ▶ Procese independente
 - ▶ nu afectează / nu e afectat de execuția altui proces

- ▶ Procese ce colaborează
 - ▶ poate afecta / fi afectat de execuția altui proces
 - ▶ necesită mecanisme de comunicare între procese
 - ▶ avantaje
 - ▶ partajare informații
 - ▶ speed-up computațional
 - ▶ modularitate



- Ce mecanisme IPC cunoașteți?

- ▶ Semafoare
 - ▶ modalitate de sincronizare

- ▶ Cozi de mesaje / Mailslots
 - ▶ comunicarea are loc prin transfer de mesaje între procesele ce colaborează

- ▶ Memorie partajată / FileMapping
 - ▶ se stabilește o zonă de memorie partajată de procesele ce cooperează
 - ▶ procesele pot schimba informații citind/scriind date din/în această zonă de memorie

- ▶ Named (sem_t)
 - ▶ Identificare prin “/nume”, inter-proces
 - ▶ sem_open(), sem_close() sem_unlink()
- ▶ Unnamed (sem_t)
 - ▶ In general inter-thread pentru acelasi proces
 - ▶ sem_init(), sem_destroy()
- ▶ P: sem_wait(), sem_trywait → EAGAIN ;)
- ▶ V: sem_post(), o unitate
- ▶ - implementare eficienta: futex(2) (Fast Userspace muTEX)
 - ▶ Evitare context-switch-uri pe anumite cazuri (care?)
- ▶ Linux: prezente ca /dev/shm/nume.sem

- ▶ Man 7 mq_overview
- ▶ Identificare prin „/nume”, tip mqd_t (intreg)
- ▶ mq_open
- ▶ mq_send - len \leq msgsize
 - ▶ msgsize - dimensiunea maximă permisă pentru un mesaj
- ▶ mq_receive - len \geq msgsize
- ▶ mq_close / mq_unlink
- ▶ Linux, maximul implicit: /proc/sys/kernel/maxmsg
- ▶ Debug: mount -t mqueue none /my/path

- ▶ Man 7 mq_overview
- ▶ shm_open
- ▶ ftruncate
 - ▶ Inițial, zona de memorie are dimensiune zero
- ▶ mmap
- ▶ munmap
- ▶ close
- ▶ shm_unlink

▶ Semafoare

- ▶ CreateSemaphore
- ▶ OpenSemaphore
- ▶ WaitForSingleObject
- ▶ ReleaseSemaphore
- ▶ CloseHandle

▶ Cozi de mesaje

`\\.\mailslot\[path]<nume>`

- ▶ CreateMailslot
- ▶ CreateFile
- ▶ ReadFile
- ▶ WriteFile
- ▶ CloseHandle

▶ Memorie partajată

- ▶ CreateFileMapping
- ▶ OpenFileMapping
- ▶ MapViewOfFile
- ▶ UnmapViewOfFile
- ▶ CloseHandle

- ▶ Procesele P1 și P2 au fiecare câte un pointer către o zonă de memorie partajată ce conține șirul "AB". Ce va conține zona de memorie după ce fiecare din procese execută:

```
for(i = 0; i < 2; i++) p[i]++;
```

- ▶ Există vreo diferență între un mutex și un semafor binar?
- ▶ Într-un sistem de planificare bazat pe priorități dați exemplu de o situație în care un task cu prioritate mare poate fi împiedicat să ruleze pentru că sistemul este ocupat să ruleze un task cu prioritate mică.