

Sistemes Operatius

Curs 2015–2016

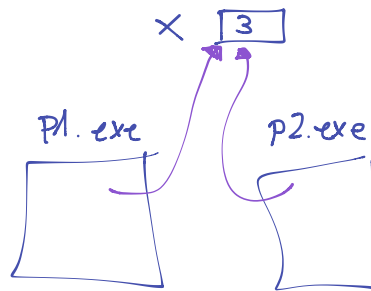
© Xavi Canaleta, 2013

4. Memòria compartida

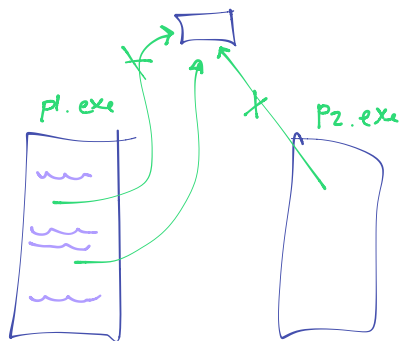
Passos:

1. Crear regió (shmget)
2. Lligar adreces (shmat)
3. Treballar
4. Deslligar adreces (shmdt)
5. Eliminar regió (shmctl)

Variable compartida →



Variable compartida →



Memoria dinamica →



ipcs → en mostra tots els recursos del sistema que unem nosaltres → MOSTRA

1. memòria compartida
 2. memòria usada
 - 3.
- La per borrar ipcrm

4. Memòria compartida

```
int main() {
    int memid;
    int *num;

    memid=shmget(0xCAFECAFE, sizeof(int), IPC_CREAT | IPC_EXCL | 0600);
    if (memid > 0) {
        num=shmat(memid, 0, 0);
        *num=0;
        while (*num != -1) {
            printf("Numero:");
            scanf("%d", num);
        }
        shmdt(num);
        shmctl(memid, IPC_RMID, NULL);
    }
    else {
        printf("Regió ja existent\n");
    }
    return 0;
}
```

demana memòria.

*si no existeix
weal.*

*si ja existeix
tornam l'ID*

< 0 ok

→ obtenir el punter.

→ treballar com una variable normal i corrent.

→ trencar punter

→ alliberar memòria.

4. Memòria compartida

```
int main() {
    int memid;
    int *num;

    memid=shmget(0xCAFECAFE, sizeof(int), IPC_EXCL|0600);
    if (memid>0) {
        num=shmat(memid, 0, 0);
        while (*num!=-1) {
            printf("Numero:%d\n", *num);
            getchar();
        }
        shmdt(num);
    }
    else {
        printf ("Regió ja existent\n");
    }
    return 0;
}
```

fer un shmget i tenim el ID de la memòria, i a partir d'aquí ja podem fer un fork per a comunicar els processos amb ID. i només ens faltaria fer el 2n procés
ja: shmatt, ja NO caldrà shmget.