

IPC Système V

3 types de communication inter-processus :

- Files de messages
- Segments de mémoire partagée
- Sémaphores

Système de nommage :

Clé codée sur 32 bits

```
ftok(char*, char)
k=ftok("/users/ens/licinf2003/gugus/fic", "A");
```

Clé spéciale pour l'héritage IPC_PRIVATE

Programme de celui qui écrit

```
int main(void)
{
    key_t k;
    struct msg m;
    int msgid, n;
    char buf[MAX];

    k=ftok("/etc/passwd", 'B');
    /* Création de la file de messages */
    msgid=msgget(k, IPC_CREAT|0666);

    while((n=read(0, buf, sizeof(buf)))>0)
    {
        m.type=1;
        m.dernier=0;
        strncpy(m.donnees, buf, n);
        /* Envoi */
        msgsnd(msgid, &n, sizeof(int)+n, 0);
    }
    m.type=1; m.dernier=0;
    msgsnd(msgid, &n, sizeof(int), 0);
    exit(0);
}
```

File de messages :

```
struct msg
{
    long type;
    int dernier;
    char donnees[MAX];
}
```

L'avantage des IPC, c'est que les 2 processus peuvent écrire dans le même tube.

Programme de celui qui lit

```
int main(void)
{
    key_t k;
    struct msg m;
    int msgid, n;

    k=ftok("/etc/passwd", 'B');
    msgid=msgget(k,0);

    do
    {
        n=msgrcv(msgid, &m, sizeof(m)-sizeof(m.type),0,0);
    }
    while(!m.dernier);

    /* Destruction de la file de message */
    msgctl(msgid, IPC_RMID, NULL);
    exit(0);
}
```

Mémoire partagée

```
/* Ecrivain */
shmkey=ftok("/etc/passwd", 'B');
shmid=shmget(shmkey, N*sizeof(int), IPC_CREAT|0666);
shmdr=shmat(shmid, NULL, 0);

t=(int*)shmdr
for(i=0; i<N; i++) t[i]=i;

/* Ecriture */
shmkey=ftok("/etc/passwd", 'B');
shmid=shmget(shmkey,0,0);
shmdr=shmat(shmid, NULL, 0);
t=(int*)shmdr;
for(i=0; i<N; i++) printf(...);
```

Synchronisation avec les sémaphores

```
P : → Puis-je ?
compteur←compteur-1
si compteur<0 alors
    attendre
fsi
```

```
V : → Vas-y !
si compteur<0 alors
    Réveiller le premier
    processus en attente
fsi
compteur←compteur+1
```

```
void P(int id)
{
static struct semops t[1]={0, -1, 0};
semop(id, t, 1);
}

void V(int id)
{
static struct semops t[1]={0, 1, 0};
semop(id, t, 1);
}

semk1=ftok("/etc/passwd", 'C');
semid1=semget(semk1, 1, IPC_CREAT|0666);
semctl(semid1, 0, IPC_SETVAL, 0);
```