

TP AISE — IO et IPCs

Exercice 1 : I/Os Structurées

Le but de cet exercice est de faire des entrées sorties binaires à partir d'une struct en C. Vous devrez stocker un tableau de struct particle de taille NUM_PART (par exemple 64) et le relire.

```
struct particle
{
    int x, y, z;
};
```

Les particules sont initialisées comme suit:

```
for( i = 0 ; i < NUM_PART; i++)
{
    parts[i].x = i;
    parts[i].y = i + 1;
}
```

Les particules sont vérifiées comme suit:

```
for( i = 0 ; i < NUM_PART; i++)
{
    if( (parts[i].x != i) || (parts[i].y != (i + 1)) )
    {
        fprintf(stderr, "Erreur dans le buffer %d\n", i);
        fprintf(stderr, "x %d vs %d\n", parts[i].x, i);
        fprintf(stderr, "y %d vs %d\n", parts[i].y, i);
        return 1;
    }
}
```

Exercice 2 : Redirection de commande

Faite un programme qui en reposant sur un pipe lance l'équivalent des commandes suivantes:

echo "8+4" | bc

Si bc n'est pas présent ou que vous ne parvenez pas à installer « bc » remplacez cette commande par « cat ».

Exercice 3 : Redirection dans un fichier

Réalisez un programme dont la sortie standard se constitue des 4096 premiers entiers. Redirigez en C cette sortie vers un fichier en ajoutant la sortie à un fichier existant si présent.

Exercice 4 : PingPong SYS-V vers POSIX

En partant de l'exemple `./exemples_IPC_sysv/file/pingpong.c` qui repose sur les files SystemV (msgrcv, msgsnd, msgctl) portez le sur les files POSIX (message queues) en utilisant les fonctions (mq_open, mq_receive, mq_send, mq_close, mq_unlink).

Exercice 5 : Segment SHM SYS-V vers POSIX

En partant de l'exemple `./exemples_IPC_sysv/shm/{shm_client.c,shm_server.c}` qui repose sur les segment de mémoire partagée SystemV (shm_*, ftruncate) portez le sur les segment de mémoire partagée POSIX (shm_open, shm_unlink).