

# TP AISE — IO et Sockets

## 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 : Client TCP

Réalisez un programme qui prendra en paramètre une adresse et un port et qui se connectera au serveur correspondant et affichera les données reçues sur la sortie standard.

Connectez vous à [towel.blinkenlights.nl](http://towel.blinkenlights.nl) sur le port 23. Que voyez vous ?

## Exercice 4 : Serveur TCP

Créez votre propre serveur TCP capable de répéter ce qui lui est envoyé au client. Vous pouvez directement le tester avec le code de l'exercice 3. Un bonus serait de gérer plusieurs clients à la fois en reposant (1) sur fork et (2) sur le multiplexage avec « select » et (3) sur les threads POSIX.