# OS308 - Système d'exploitation


## TP7 - Communication par tube

Auteurs : CASTANIER Raphaël, RIUS BERNAL Joffrey
Date    : 11/06/2018

## Exercice 1

```c
// OS308   : Système d'exploitation
// TP7     : Communication par tube

// Auteur  : CASTANIER Raphaël, RIUS BERNAL Joffrey
// Date    : 07/06/2018
// Compil. : gcc exo1.c -o prog -Wall -Werror -Wextra --std=c99
// Exercice 1 - Tube ordinaire


#include <sys/types.h>  // pid_t
#include <unistd.h>     // pipe, fork, read, write
#include <string.h>     // strlen
#include <stdio.h>      // fprintf, getc
#include <stdlib.h>     // exit

#define VAL_MAX 1024    // taille du buffer

int main (void)
{
    // creation du pipe
    int p1[2];
    int p2[2];
    if (pipe(p1) == -1) goto error;
    if (pipe(p2) == -1) goto error;

    // descripteurs
    int readDescriptor1  = p1[0];
    int writeDescriptor1 = p1[1];
```

```
    int readDescriptor2  = p2[0];
    int writeDescriptor2 = p2[1];

    // fork
    pid_t fils1 = fork();
    if (fils1 == -1) goto error;

    // Fils1
    if (fils1 == 0)
    {
        // fermeture des descripteurs inutilisés
        close(readDescriptor1);
        close(readDescriptor2);
        close(writeDescriptor2);

        char* message1 = "Je suis le premier fils";

        // +1 pour le caractère de fin de chaine
        write(writeDescriptor1, message1, strlen(message1)+1);

        // fermeture par le fils du descripteur d'ecriture
        close(writeDescriptor1);
    }

    // Pere
    if (fils1 != 0)
    {
        // fork
        pid_t fils2 = fork();
        if (fils2 == -1) goto error;

        // Fils2
        if (fils2 == 0)
        {
            // fermeture des descripteurs inutilisés
            close(readDescriptor2);
            close(readDescriptor1);
            close(writeDescriptor1);

            char* message2 = "Je suis le deuxieme fils";

            // +1 pour le caractère de fin de chaine
            write(writeDescriptor2, message2, strlen(message2)+1);

            // fermeture par le fils du descripteur d'ecriture
            close(writeDescriptor2);
        }

        // Pere
        else
        {
            // fermeture des descripteurs inutilisés
            close(writeDescriptor1);
            close(writeDescriptor2);
```

```
            // buffers
            char contenu1[VAL_MAX];
            char contenu2[VAL_MAX];

            // lecture de tubes et gestion d'erreur
            if (read(readDescriptor1, contenu1, VAL_MAX) == 0) goto error;
            if (read(readDescriptor2, contenu2, VAL_MAX) == 0) goto error;

            // affichage du contenu lu
            printf("%s\n", contenu1);
            printf("%s\n", contenu2);

            // fermeture des descripteurs de lecture
            close(readDescriptor1);
            close(readDescriptor2);
        }
    }

    return EXIT_SUCCESS;

    // gestion de la fermeture des tubes en cas d'erreur
    error:
        close(writeDescriptor1);
        close(writeDescriptor2);
        close(readDescriptor1);
        close(readDescriptor2);
        exit(EXIT_FAILURE);
}
```

# Exercice 2

## Client

```
// OS308   : Système d'exploitation
// TP7     : Communication par tube
// Auteur  : CASTANIER Raphaël, RIUS BERNAL Joffrey
// Date    : 11/06/2018
// Compil. : gcc client.c -o client -Wall -Werror -Wextra --std=c99
// Exercice 2 - Tube Client/Serveur

#include <stdio.h>   // fopen, fputs
#include <stdlib.h>  // exit

#define PATH "/tmp/fifo" // nom du tube nommé (path)
#define VAL_MAX 1024     // taille du buffer

int main (int argc, char const *argv[])
{
    // ouverture du tube
    FILE *f = fopen(PATH, "w");

    // gestion d'erreur
```

```
    if (f == NULL)
    {
        exit(EXIT_FAILURE);
    }
    else
    {
        // ecriture des arguements dans le tube
        for (int i=1; i<argc; i++)
        {
            if (fputs(argv[i], f) == EOF)
            {
                printf("Erreur d'ecriture\n");
                exit(EXIT_FAILURE);
            }
            fputs(" ", f); // ajout d'un espace entre chaque argument
        }
    }

    return (EXIT_SUCCESS);
}
```

## Serveur

```
// OS308   : Système d'exploitation
// TP7     : Communication par tube
// Auteur  : CASTANIER Raphaël, RIUS BERNAL Joffrey
// Date    : 11/06/2018
// Compil. : gcc serveur.c -o serveur -Wall -Werror -Wextra --std=c99

// Exercice 2 - Tube Client/Serveur

#include <stdio.h>     // fprintf
#include <stdlib.h>    // exit
#include <errno.h>     // errno
#include <unistd.h>    // unlink
#include <sys/types.h> // mkfifo
#include <sys/stat.h>  // mkfifo
#include <string.h>    // strerror

#define VAL_MAX 1024     // taille du buffer
#define PATH "/tmp/fifo" // nom du tube nommé (path)

int main (int argc, char const *argv[])
{
    // utilisation factice des arguments
    (void)argc;
    (void)argv;

    // suppression du tube s'il existe
    unlink(PATH);

    // création du tube nommé
    if (mkfifo(PATH, 0666) != 0) // droits d'accès : rw-rw-rw-
```

```c
    {
        fprintf(stderr, "Impossible de creer le tube nommé\n");
        fprintf(stderr, "Erreur : %s\n", strerror(errno));
        exit(EXIT_FAILURE);
    }

    char buf[VAL_MAX]; // créatuon du buffer
    FILE *f = fopen(PATH, "r"); // ouverture du tube

    // lecture du tube
    while (1)
    {
        // si la lecture du tube change de contenu, afficher
        if (fgets(buf, VAL_MAX, f) != NULL)
        {
            printf("%s \n", buf);
        }
    }

    return (EXIT_SUCCESS);
}
```