

```
/* Gestion des processus sous Linux */

/* TDM1 - Exercice 1 */

#include <sys/types.h>
#include <unistd.h>
#include <stdlib.h>
#include <stdio.h>

void traitFils1();
void traitFils2();

int main()
{
    pid_t idProc ;
    int rapport, numSignal, statut;

    /* processus pere */
    /* creation du fils1 */
    idProc = fork() ;
    switch (idProc )
    {
        case -1 : perror("echec fork");exit(1); /* erreur */

        case 0 : traitFils1(); exit(1) ; /* appel du traitement du fils1 */
    }

    /* creation du fils2 */
    idProc = fork() ;
    switch (idProc )
    {
        case -1 : perror("echec fork");exit(1); /* erreur */

        case 0 : traitFils2(); exit(1) ; /* appel du traitement du fils2 */
    }

    /* suite du processus pere */
    /* attente de la terminaison des fils */

    idProc = wait( &rapport ) ;
    while ( idProc != -1 )
    {
        printf("\nTerminaison du fils de PID= %d\n", idProc);

        numSignal = rapport & 0x7F ;
        switch ( numSignal )
        {
            case 0 : /* fin normale */
                statut = (rapport >>8)& 0xFF ;
                printf("Fin normale, statut= %d\n", statut); break ;

            default : /* fin anormale */
                printf("Fin anormale, numSignal= %d\n", numSignal);
        }
        idProc = wait( &rapport ) ;
    }
    /* fin du processus pere */

    /* traitement du fils1 */
    void traitFils1()
    {
        printf("\n***fils1 --> PID= %d\n", getpid());
        exit(3);
    }

    /* traitement du fils2 */
    void traitFils2()
    {
        int v=10;
```

```

printf("\n***fils2 --> PID= %d\n", getpid());
*(int *)NULL=v;
}

/* Gestion des processus sous Linux */

/* TDM1 - Exercice 2 */

#include <sys/types.h>
#include <unistd.h>
#include <stdlib.h>
#include <stdio.h>

int main()
{
    pid_t idProc ;
    int err,rapport, numSignal, statut;

    char * tabParam[3]={"p2fils2", "123456", NULL};

    /* processus pere */
    /* creation du fils1 */
    idProc = fork() ;
    switch (idProc )
    {
        case -1 : perror("echec fork");exit(1); /* erreur */

        case 0 : /* execution du traitement du fils1 */
            err=execl("./p2fils1", "p2fils1", "coucou", NULL);
            if (err == -1)
                { perror("echec execl");exit(2);}
    }

    /* creation du fils2 */
    idProc = fork() ;
    switch (idProc )
    {
        case -1 : perror("echec fork");exit(1); /* erreur */

        case 0 : /* execution du traitement du fils2 */
            err=execvp("p2fils2", tabParam);
            if (err == -1)
                { perror("echec execvp");exit(3);}
    }

    /* suite du processus pere */
    /* attente de la terminaison des fils */

    idProc = wait( &rapport ) ;
    while ( idProc != -1 )
    {
        printf("\nTerminaison du fils de PID= %d\n", idProc);

        numSignal = rapport & 0x7F ;
        switch ( numSignal )
        {
            case 0 : /* fin normale */
                statut = ( rapport >> 8 ) & 0xFF ;
                printf("Fin normale, statut= %d\n", statut); break ;

            default : /* fin anormale */
                printf("Fin anormale, numSignal= %d\n", numSignal);
        }
        idProc = wait( &rapport ) ;
    }
}

```

```
}
/* fin du processus pere */

/* Gestion des processus sous Linux */

/* TDM1 - Exercice 2 */

/* traitement du fils1 */

#include <sys/types.h>
#include <unistd.h>
#include <stdlib.h>
#include <stdio.h>
#include <string.h>

int main(int nbParam, char * tabParam[])
{
    char chaine[50+1];

    switch (nbParam)
    {
        case 2 : /* recuperation du parametre 1 */
                    strncpy(chaine, tabParam[1],50);
                    chaine[50]='\0';
                    break;

        default : printf("***fils1 --> nombre de parametres incorrect!!!");
                    exit(1);
    }

    printf("\n***fils1 --> PID= %d\n", getpid());
    printf("***fils1 -->tabParam[1]= %s\n", tabParam[1]);
    exit(3);
}

/* Gestion des processus sous Linux */

/* TDM1 - Exercice 2 */

/* traitement du fils2 */

#include <sys/types.h>
#include <unistd.h>
#include <stdlib.h>
#include <stdio.h>

int main(int nbParam, char * tabParam[])
{
    int v=10,nbre;

    switch (nbParam)
    {
        case 2 : /* conversion du parametre 1 */
                    sscanf(tabParam[1], "%d",&nbre);
                    break;

        default : printf("***fils2 --> nombre de parametres incorrect!!!");
                    exit(1);
    }

    printf("\n***fils2 --> PID= %d\n", getpid());
    printf("***fils2 --> tabParam[1]= %d\n", nbre);

    *(int *)NULL=v;
}
```