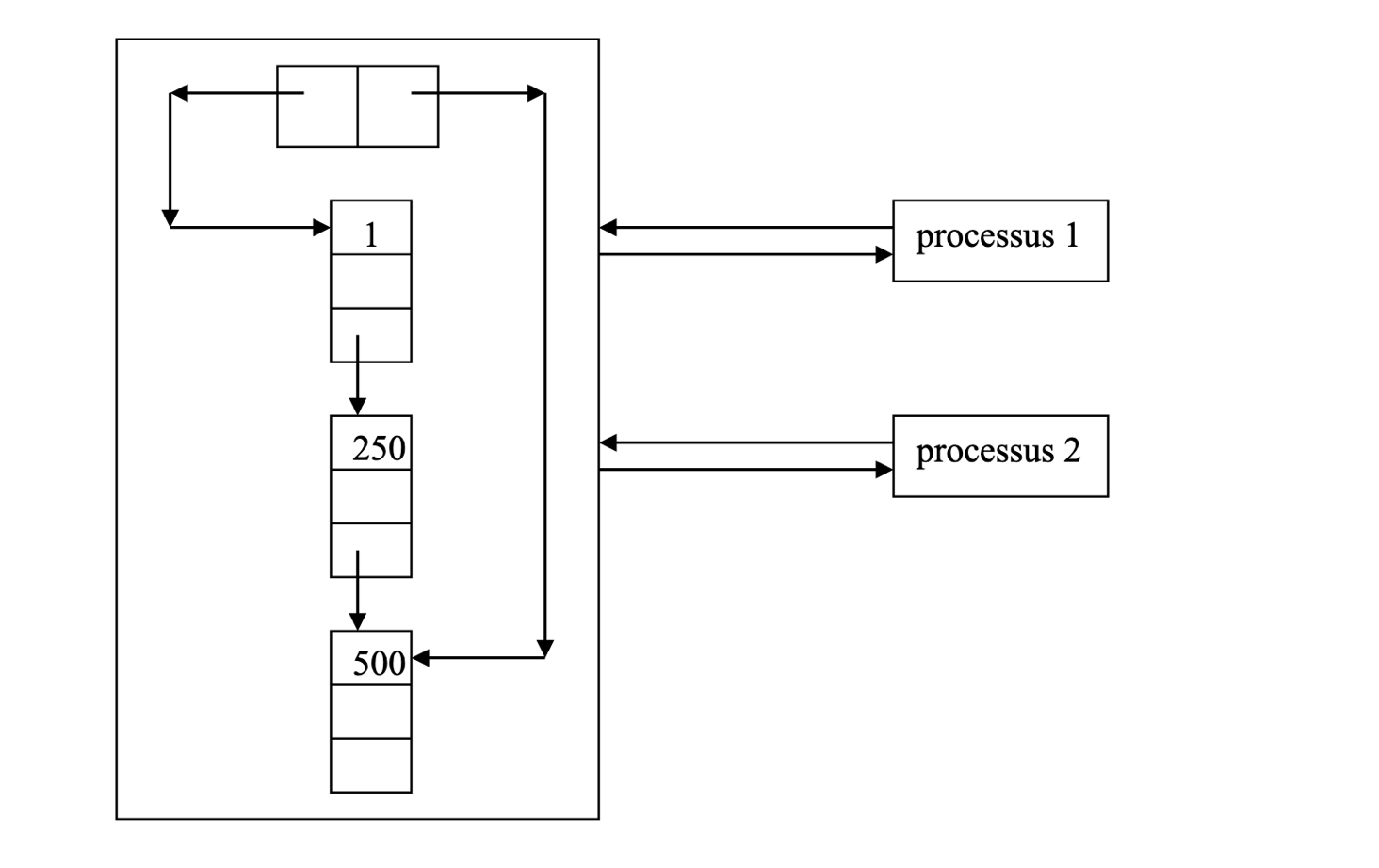
Travaux Pratique N° 4

But :

Création d’une file de message permet d'échanger des messages entre processus.



Réalisé par : Khallouk Achraf.

**Exercice 1**

1. Ecrire un programme fm1.c qui crée une file de messages et envoie un message “test IPC” à la file. Le mtype est un entier quelconque.

Pour Créer une file de message on utilise l’appel système ‘msgget’ qui permet de créer la file de message de nom cle (dans notre exemple) si celle-ci n'existe pas encore; si la file existe déjà, et permet d'obtenir l'identificateur de la file, que le processus utilisera par la suite pour manipuler la file de messages.

Et on utilise l’appel syteme ‘msgsnd’ qui nous alloue d’envoi des message à la file

Code Source de fm1.c :

#include<stdio.h>

#include<sys/msg.h>

#include<unistd.h>

#include<time.h>

#include<stdlib.h>

main(){

/\* (cours) Structure pour envoyer ou recevoir les messages \*/

typedef struct{

long mtype;

char mtext[256]; }MESSAGE;

MESSAGE message;

#define LGMES sizeof(message.mtext)

key\_t cle;

int numfms=-1;

//creation de la file

printf("\n\n++++++++++++++++Creation de la fil++++++++++++++++++++++\n\n");

printf("introduire la cle: ");

scanf("%d",&cle);

numfms=msgget(cle,IPC\_CREAT | 0666);

if(numfms==-1){perror("case 1 erreur msgget"); exit(1);}

printf("Num interne: %d\n",numfms);

//envoi de message

printf("\n\n++++++++++++++++Envoi de la message++++++++++++++++++++++\n\n");

printf("entrer le PID de destinataire : ");

scanf("%d",&message.mtype);

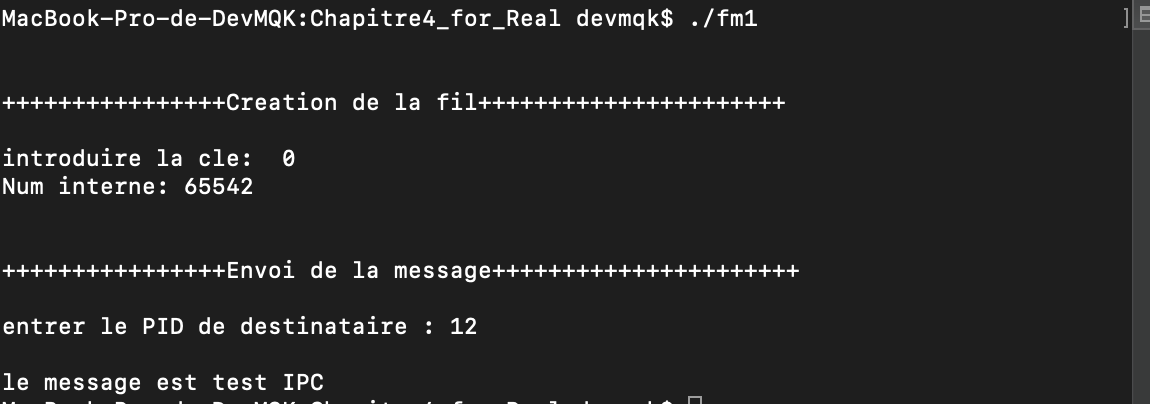
sprintf(message.mtext,"test IPC");

printf("\nle message est %s\n","test IPC");

if(msgsnd(numfms,&message,LGMES,IPC\_NOWAIT)==-1){perror("case 4 erreur msgsnd");}

}

Test :



1. Ecrire un programme fm2.c qui consulte la file de messages créée par fm1.c et reçoit le message envoyé. Le programme fm2.c affiche le nombre de message de la  
     
   Pour lire le message envoyer par fm1.c il faut utiliser l’appele systeme ‘msgrcv’ qui besoin de l’identificateur de la file de messages.

Code Source:

#include<stdio.h>

#include<sys/msg.h>

#include<unistd.h>

#include<time.h>

#include<stdlib.h>

main(){

/\* (cours) Structure pour envoyer ou recevoir les messages \*/

typedef struct{

long mtype;

char mtext[256]; }MESSAGE;

MESSAGE message;

#define LGMES sizeof(message.mtext)

key\_t cle;

int n;

int numfms;

printf("saisir le Num interne generer : ");

scanf("%d",&numfms);

n=msgrcv(numfms,&message,LGMES,0,IPC\_NOWAIT);

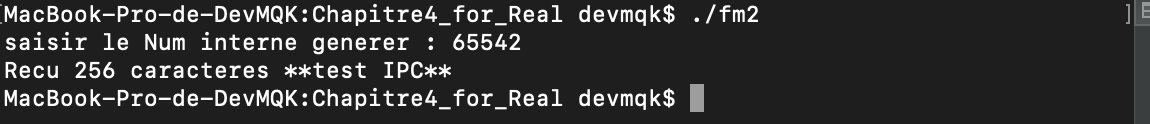
if(n==-1){perror("case 5 erreur msgrcv");

exit(1);}

else printf("Recu %d caracteres \*\*%s\*\*\n",n,message.mtext);

}

Test :



Exercice 2

Modifier les programmes de l’Exercice 1 pour que fm1.c soit exécuté après réception du

signal SIGINT et que fm2.c soit lancé après réception du signal SIGTSTP.

Dans cette exercice il faut juste ajouter les signals SIGINT et SIGTSTP pour que le processus attend SIGINT (CTRL+C) pour créer la file de message et envoyer un message a la file, et le deuxième processus attend SIGTSTP (CTRL+Z) pour lire de la file.

Code source de fm1 (modifié):  
#include<stdio.h>

#include<sys/msg.h>

#include<unistd.h>

#include<time.h>

#include<stdlib.h>

void envoi(){

/\* (cours) Structure pour envoyer les messages \*/

typedef struct{

long mtype;

char mtext[256]; }MESSAGE;

MESSAGE message;

#define LGMES sizeof(message.mtext)

key\_t cle;

int numfms=-1;

//creation de la file

printf("\n\n++++++++++++++++Creation de la fil++++++++++++++++++++++\n\n");

printf("introduire la cle: ");

scanf("%d",&cle);

numfms=msgget(cle,IPC\_CREAT | 0666);

if(numfms==-1){perror("case 1 erreur msgget"); exit(1);}

printf("Num interne: %d\n",numfms);

//envoi de message

printf("\n\n++++++++++++++++Envoi de la message++++++++++++++++++++++\n\n");

printf("entrer le PID de destinataire : ");

scanf("%d",&message.mtype);

sprintf(message.mtext,"test IPC");

printf("\nle message est %s\n","test IPC");

if(msgsnd(numfms,&message,LGMES,IPC\_NOWAIT)==-1){perror("case 4 erreur msgsnd");}

}

main(){

signal(SIGINT,envoi);

for(;;);

}

Test :



Code source de fm2 (modifié):

#include<stdio.h>

#include<sys/msg.h>

#include<unistd.h>

#include<time.h>

#include<stdlib.h>

void recoi(){

/\* (cours) Structure pour recevoir les messages \*/

typedef struct{

long mtype;

char mtext[256]; }MESSAGE;

MESSAGE message;

#define LGMES sizeof(message.mtext)

key\_t cle;

int n;

int numfms;

printf("saisir le Num interne generer : ");

scanf("%d",&numfms);

n=msgrcv(numfms,&message,LGMES,0,IPC\_NOWAIT);

if(n==-1){perror("case 5 erreur msgrcv");

exit(1);}

else printf("Recu %d caracteres \*\*%s\*\*\n",n,message.mtext);

}

main(){

signal(SIGTSTP,recoi);

for(;;);

}

TEST :