Họ và tên: Nguyễn Trọng Đạt

MSSV: 52100176

Lớp: 21050301

Lab 5\_3

Bài 1:

Share Memory:

#include <stdio.h>

#include <unistd.h>

#include <limits.h>

#include <string.h>

#include <stdlib.h>

#include <sys/types.h>

#include <sys/ipc.h>

#include <sys/shm.h>

#define SIZE 256

int main(int argc, char\* argv[])

{

int \*shm, shmid, k,pid;

key\_t key;

if((key=ftok(".",65))==-1){

perror("Key created.\n");

return 1;

}

shmid = shmget(key, SIZE, IPC\_CREAT | 0666);

if (shmid == -1) {

perror("Shared memory created.\n");

return 2;

}

shm = (int\*) shmat(shmid, 0, 0);

pid = fork();

if(pid==0) { // child

shm[0] = atoi(argv[1]);

sleep(4);

printf ("%d!= %d\n", shm[0],shm[1]);

shmdt((void\*) shm);

shmctl(shmid, IPC\_RMID, (struct shmid\_ds\*) 0);

return 0;

}

else if(pid >0) { // parent

sleep(2);

int i,cnt=1;

for(i=1;i<=shm[0];i++){

cnt\*=i;

}

shm[1]=cnt;

shmdt((void\*) shm);

sleep(5);

return 0;

}

else { perror("Fork failed."); return 4; }

return 0;

}

Text

Description automatically generated

Message queue:

Write.c

Text

Description automatically generated

Read.c

Text

Description automatically generated

Text

Description automatically generated

Bài 2:

Share Memory:

#include <stdio.h>

#include <unistd.h>

#include <limits.h>

#include <string.h>

#include <stdlib.h>

#include <sys/types.h>

#include <sys/ipc.h>

#include <sys/shm.h>

#define SIZE 256

int main(int argc, char \*argv[])

{

int \*shm, shmid, k, pid;

key\_t key;

if ((key = ftok(".", 65)) == -1)

{

perror("Key created.\n");

return 1;

}

shmid = shmget(key, SIZE, IPC\_CREAT | 0666);

if (shmid == -1)

{

perror("Shared memory created.\n");

return 2;

}

shm = (int \*)shmat(shmid, 0, 0);

pid = fork();

if (pid == 0)

{ // child

shm[0] = atoi(argv[1]);

shm[1] = atoi(argv[2]);

shm[2] = (int)(argv[3][0]);

sleep(3);

switch (shm[2])

{

case 43:

printf("%d+%d=%d\n", shm[0],shm[1],shm[3]);

break;

case 45:

printf("%d-%d=%d\n", shm[0],shm[1],shm[3]);

break;

case 120:

printf("%d\*%d=%d\n", shm[0],shm[1],shm[3]);

break;

case 47:

printf("%d/%d=%d\n", shm[0],shm[1],shm[3]);

break;

}

shmdt((void \*)shm);

shmctl(shmid, IPC\_RMID, (struct shmid\_ds \*)0);

return 0;

}

else if (pid > 0)

{ // parent

printf("Data %d",shm[2]);

sleep(1);

if(shm[2]==43){

shm[3]=shm[1]+shm[0];

}else if(shm[2]==45){

shm[3]=shm[1]-shm[0];

}else if(shm[2]==120){

shm[3]=shm[1]\*shm[0];

}else if(shm[2]==47){

shm[3]=shm[0]\*1.0/shm[1];

}

shmdt((void \*)shm);

sleep(5);

return 0;

}

else

{

perror("Fork failed.");

return 4;

}

return 0;

}

Text

Description automatically generated

Message queue:

Write.c

Graphical user interface, text, application

Description automatically generated

Read.c

Text

Description automatically generated

Text

Description automatically generated with medium confidence

Text

Description automatically generated

Bài thêm:

Share Memory:

#include <stdio.h>

#include <unistd.h>

#include <limits.h>

#include <string.h>

#include <stdlib.h>

#include <sys/types.h>

#include <sys/ipc.h>

#include <sys/shm.h>

#include<time.h>

#define SIZE 256

int main(int argc, char\* argv[])

{

srand(time(NULL));

int i;

int \*shm, shmid, k,pid;

key\_t key;

if((key=ftok(".",65))==-1){

perror("Key created.\n");

return 1;

}

shmid = shmget(key, SIZE, IPC\_CREAT | 0666);

if (shmid == -1) {

perror("Shared memory created.\n");

return 2;

}

shm = (int\*) shmat(shmid, 0, 0);

pid = fork();

if(pid==0) { // child

FILE \*f = fopen("data", "w");

int n = atoi(argv[1]);

for (i = 0; i < n; ++i)

{

fprintf(f,"%d\n", rand() % 100);

}

fclose(f);

FILE \*f1 = fopen("data","r");

int k,x=1;

shm[0]=n;

shm[shm[0]+1]=-1;

while(fscanf(f1,"%d",&k) != EOF)

{

shm[x] = k;

x++;

}

fclose(f1);

//

sleep(3);

printf("Sum=%d\n",shm[shm[0]+1]);

printf("Mang sau khi sap xep:\n");

for (i = 1; i <= shm[0]; ++i)

{

printf("%d ",shm[i]);

}

shmdt((void\*) shm);

shmctl(shmid, IPC\_RMID, (struct shmid\_ds\*) 0);

return 0;

}

else if(pid >0) { // parent

sleep(1);

int sum=0;

for (i = 1; i <= shm[0]; ++i)

{

sum+=shm[i];

}

shm[shm[0]+1]=sum;

//sort

int j,k;

for (i = 1; i < shm[0]; ++i)

{

for (j = 1; j < shm[0]; ++j)

{

if (shm[i] < shm[j])

{

k = shm[i];

shm[i] = shm[j];

shm[j] = k;

}

}

}

//TODO

shmdt((void\*) shm);

sleep(5);

return 0;

}

else { perror("Fork failed."); return 4; }

return 0;

}

A screenshot of a computer

Description automatically generated with medium confidence

Message queue:

Write.c

Text

Description automatically generated

Read.c

Text

Description automatically generated

Text

Description automatically generated