

Huỳnh Tấn Dương 3122410061

Hoàn thành 3/3

Bài 1/cách fork());

```
1  #include<stdio.h>
2  #include<sys/ipc.h>
3  #include<sys/msg.h>
4  #include<unistd.h>
5
6  struct mesg_buffer{
7      long mesg_type;
8      int mesg_text;
9  }message;
10
11 int giaithua(int n){
12     if(n==1)return 1;
13     return n*giaithua(n-1);}
14
15 int main(int argc,char*argv[]){
16     key_t key;
17     key=ftok(".",1);
18     int n=atoi(argv[1]);
19     int msgid;
20
21     int id =fork();
22     if(id>0){ //parent
23         msgid=msgget(key,0666|IPC_CREAT);
24         message.mesg_type=1;
25         message.mesg_text=giaithua(n);
26         msgsnd(msgid,&message,sizeof(message),0);
27         printf("giai thua cua n dc gui di la %d\n",message.mesg_text);
28         wait(NULL);
29     }
30     else if(id==0){ //child
31         msgid=msgget(key,0666|IPC_CREAT);//khoei tao msgid trong tung tien trinh
32         msgrcv(msgid,&message,sizeof(message),1,0);
33         printf("nhan duoc giai thua cua n la %d\n",message.mesg_text);
34         msgctl(msgid,IPC_RMID,NULL);
35     }
36     else
37         printf("failed to fork\n");
38     return 0;
39 }
40
```

```
duong@ubuntu: ~/Desktop/baitap/lab9/bai1/cach1
duong@ubuntu:~/Desktop/baitap/lab9/bai1/cach1$ ./bai1 4
giai thua cua n dc gui di la 24
nhan duoc giai thua cua n la 24
duong@ubuntu:~/Desktop/baitap/lab9/bai1/cach1$
```

Cách 2 file riêng biệt

```
part1.c
1 #include<stdio.h>
2 #include<sys/ipc.h>
3 #include<sys/msg.h>
4 struct mesagge_buffer{
5     long mesg_type;
6     int mesg_text;
7 }message;
8
9 int main(int argc, char*argv[]){
10     key_t key;
11     int n=atoi(argv[1]);
12     key=ftok(".",1);
13     int msgid=msgget(key,0666|IPC_CREAT);
14     message.mesg_text=n;
15     message.mesg_type=1;
16     msgsnd(msgid,&message,sizeof(message),0);
17     printf("du lieu dc gui di:%d\n",message.mesg_text);
18     return 0;
19 }
20

part2.c
1 #include<stdio.h>
2 #include<sys/ipc.h>
3 #include<sys/msg.h>
4 struct message_buffer{
5     long msg_type;
6     int msg_text;
7 }message;
8
9 int giaithua(int n){
10     if(n==1)return 1;
11     return n*giaithua(n-1);
12 }
13 int main(int agrc, char *argv[]){
14     key_t key=ftok(".",1);
15     int msgid=msgget(key,0666|IPC_CREAT);
16     msgrcv(msgid,&message,sizeof(message),1,0);
17     printf("nhan duoc %d\n",message.msg_text);
18     printf("giai thua cua %d la %d\n",message.msg_text,giaithua(message.msg_text));
19
20     return 0;
21 }
22

duong@ubuntu: ~/Desktop/baitap/lab9/bai1/cach2
duong@ubuntu:~$ cd Desktop/baitap/lab9/bai1/cach2
duong@ubuntu:~/Desktop/baitap/lab9/bai1/cach2$ gcc -c part1.c
duong@ubuntu:~/Desktop/baitap/lab9/bai1/cach2$ gcc -o part1 part1.o
duong@ubuntu:~/Desktop/baitap/lab9/bai1/cach2$ ./part1 4
du lieu dc gui di:4
duong@ubuntu:~/Desktop/baitap/lab9/bai1/cach2$
duong@ubuntu:~/Desktop/baitap/lab9/bai1/cach2$ ./part1 4
du lieu dc gui di:4
duong@ubuntu:~/Desktop/baitap/lab9/bai1/cach2$

duong@ubuntu:~/Desktop/baitap/lab9/bai1/cach2$ gcc -c part2.c
duong@ubuntu:~/Desktop/baitap/lab9/bai1/cach2$ gcc -o part2 part2.o
duong@ubuntu:~/Desktop/baitap/lab9/bai1/cach2$ ./part2
nhan duoc 4
giai thua cua 4 la 24
duong@ubuntu:~/Desktop/baitap/lab9/bai1/cach2$ ./part2
nhan duoc 4
giai thua cua 4 la 24
duong@ubuntu:~/Desktop/baitap/lab9/bai1/cach2$ ./part2
nhan duoc 4
giai thua cua 4 la 24
duong@ubuntu:~/Desktop/baitap/lab9/bai1/cach2$
```

Bài 2/cách fork;

```
1  #include<sys/ipc.h>
2  #include<sys/msg.h>
3  #include<unistd.h>
4  #include<stdio.h>
5  #include<string.h>
6  struct message_buffer{
7      long msg_type;
8      int msg_data[3];
9      char msg_sign[5];
10 }message;
11 int main(int argc, char*argv[]){
12     key_t key1=ftok(".",1);
13     key_t key2=ftok(".",1);
14     int msgid1,msgid2;
15     int x1,x2;
16     int chldid=fork();
17
18     if(chldid>0){//parent
19         msgid1=msgget(key1,0666|IPC_CREAT);//gui tu parent sang chld
20         msgid2=msgget(key2,0666|IPC_CREAT);//gui tu chld ve parent
21         message.msg_type=1;
22         message.msg_data[0]=atoi(argv[1]);
23         message.msg_data[1]=atoi(argv[2]);
24         strcpy(message.msg_sign,argv[3]);
25         msgsnd(msgid1,&message,sizeof(message),0);
26
27         printf("data da gui la %d\n",message.msg_data[0]);
28         printf("data da gui la %d\n",message.msg_data[1]);
29         printf("data da gui la %s\n",message.msg_sign);
30     }
```

```

30
31     msgrcv(msgid2,&message,sizeof(message),2,0);
32     printf("ket qua la %d\n",message.msg_data[2]);
33     msgctl(msgid1,IPC_RMID,NULL);
34 }
35 else if(childdid==0){//child
36     int kq;char sign[5];
37     msgid1=msgget(key1,0666|IPC_CREAT);
38     msgid2=msgget(key2,0666|IPC_CREAT);
39     msgrcv(msgid1,&message,sizeof(message),1,0);
40
41     x1=message.msg_data[0];
42     x2=message.msg_data[1];
43     strcpy(sign,message.msg_sign);
44     printf("data nhan duoc %d %d %s\n",x1,x2,sign);
45
46     if(strcmp(sign,"+")==0)kq=x1+x2;
47     else if(strcmp(sign,"-")==0)kq=x1-x2;
48     else if(strcmp(sign,"x")==0)kq=x1*x2;
49     else if(strcmp(sign,"/")==0)kq=x1/x2;
50     else printf("loi phep tinh\n");
51
52     message.msg_data[2]=kq;
53     message.msg_type=2;
54     msgsnd(msgid2,&message,sizeof(message),0);
55 }
56 return 0;
57 }

```

```

duong@ubuntu:~/Desktop/baitap/lab9/bai2/cach1$ gcc -c bai2.c
duong@ubuntu:~/Desktop/baitap/lab9/bai2/cach1$ gcc -o bai2 bai2.o
duong@ubuntu:~/Desktop/baitap/lab9/bai2/cach1$ ./bai2 6 4 +
data da gui la 6
data da gui la 4
data da gui la +
data nhan duoc 6 4 +
ket qua la 10
duong@ubuntu:~/Desktop/baitap/lab9/bai2/cach1$ ./bai2 6 4 -
data da gui la 6
data da gui la 4
data da gui la -
data nhan duoc 6 4 -
ket qua la 2
duong@ubuntu:~/Desktop/baitap/lab9/bai2/cach1$ ./bai2 6 4 x
data da gui la 6
data da gui la 4
data da gui la x
data nhan duoc 6 4 x
ket qua la 24
duong@ubuntu:~/Desktop/baitap/lab9/bai2/cach1$ ./bai2 4 2 /
data da gui la 4
data da gui la 2
data da gui la /
data nhan duoc 4 2 /
ket qua la 2
duong@ubuntu:~/Desktop/baitap/lab9/bai2/cach1$

```

Cách 2 file riêng biệt

```
part1.c
1 #include<sys/ipc.h>
2 #include<sys/msg.h>
3 #include<unistd.h>
4 #include<stdio.h>
5 #include<string.h>
6 struct message_buffer{
7     long msg_type;
8     int msg_data[3];
9     char sign[5];
10 }message;
11 int main(int argc,char*argv[]){
12     key_t key1=ftok("k1",1);//1 la ky tu dinh danh project
13     key_t key2=ftok("k2",2);//2 la ky tu dinh danh project
14
15     int msgid1=msgget(key1,0666|IPC_CREAT);//msg gui tu part1->part2,gui input sang part2
16     int msgid2=msgget(key2,0666|IPC_CREAT);//msg gui tu part2->part1,gui output sang part1
17
18     message.msg_data[0]=atoi(argv[1]);
19     message.msg_data[1]=atoi(argv[2]);
20     strcpy(message.sign,argv[3]);
21     printf("Data da gui la %d %d %s\n",message.msg_data[0],message.msg_data[1],message.sign);
22
23     message.msg_type=1;
24     msgsnd(msgid1,&message,sizeof(message),0);
25     msgrcv(msgid2,&message,sizeof(message),2,0);//doi so thu 4 la msg_type
26
27     printf("Ket qua la %d\n",message.msg_data[2]);
28     msgctl(msgid2,IPC_RMID,NULL);
29     return 0;
30 }
```

```
part2.c
4 #include<stdio.h>
5 #include<string.h>
6 struct message_buffer{
7     long msg_type;
8     int msg_data[3];
9     char sign[5];
10 }message;
11 int main(int argc,char*argv[]){
12     char sign[5];
13     key_t key1=ftok("k1",1);
14     key_t key2=ftok("k2",2);
15     int msgid1=msgget(key1,0666|IPC_CREAT);
16     int msgid2=msgget(key2,0666|IPC_CREAT);
17
18     msgrcv(msgid1,&message,sizeof(message),1,0);
19     strcpy(sign,message.sign);
20
21     printf("Data nhan duoc %d %d %s\n",message.msg_data[0],message.msg_data[1],message.sign);
22
23     if(strcmp(message.sign,"+")==0)message.msg_data[2]=message.msg_data[0]+message.msg_data[1];
24     else if(strcmp(message.sign,"-")==0)message.msg_data[2]=message.msg_data[0]-message.msg_data[1];
25     else if(strcmp(message.sign,"x")==0)message.msg_data[2]=message.msg_data[0]*message.msg_data[1];
26     else if(strcmp(message.sign,"/")==0)message.msg_data[2]=message.msg_data[0]/message.msg_data[1];
27     else printf("Loi phap tinh\n");
28
29     message.msg_type=2;
30     msgsnd(msgid2,&message,sizeof(message),0);
31     msgctl(msgid1,IPC_RMID,NULL);
32     return 0;
33 }
```

```
# duong@ubuntu: ~/Desktop/baitap/lab9/bai2/cach2
#duong@ubuntu:~/Desktop/baitap/lab9/bai2/cach2$ gcc -c part1.c
duong@ubuntu:~/Desktop/baitap/lab9/bai2/cach2$ gcc -o part1 part1.o
duong@ubuntu:~/Desktop/baitap/lab9/bai2/cach2$ ./part1 4 6 +
Data da gui la 4 6 +
Ket qua la 10
duong@ubuntu:~/Desktop/baitap/lab9/bai2/cach2$ ./part1 6 4 -
Data da gui la 6 4 -
Ket qua la 2
duong@ubuntu:~/Desktop/baitap/lab9/bai2/cach2$ ./part1 6 4 x
Data da gui la 6 4 x
Ket qua la 24
duong@ubuntu:~/Desktop/baitap/lab9/bai2/cach2$ ./part1 10 2 /
Data da gui la 10 2 /
Ket qua la 5
duong@ubuntu:~/Desktop/baitap/lab9/bai2/cach2$
```

```
duong@ubuntu: ~/Desktop/baitap/lab9/bai2/cach2
duong@ubuntu:~/Desktop/baitap/lab9/bai2/cach2$ gcc -c part2.c
duong@ubuntu:~/Desktop/baitap/lab9/bai2/cach2$ gcc -o part2 part2.o
duong@ubuntu:~/Desktop/baitap/lab9/bai2/cach2$ ./part2
Data nhan duoc 4 6 +
duong@ubuntu:~/Desktop/baitap/lab9/bai2/cach2$ ./part2
Data nhan duoc 6 4 -
duong@ubuntu:~/Desktop/baitap/lab9/bai2/cach2$ ./part2
Data nhan duoc 6 4 x
duong@ubuntu:~/Desktop/baitap/lab9/bai2/cach2$ ./part2
Data nhan duoc 10 2 /
duong@ubuntu:~/Desktop/baitap/lab9/bai2/cach2$
```

Bài 3/

```
duong@ubuntu: ~/Desktop/test1
1 2 3 4 5 15
3 4 5 12
1 6 7 9 23
3 4 5 6 7 8 33
1 2 3 4 5 15
duong@ubuntu:~/Desktop/test1$ ./p1
duong@ubuntu:~/Desktop/test1$ ./p1
duong@ubuntu:~/Desktop/test1$ ./p1
3 4 5 12
1 6 7 9 23
3 4 5 6 7 8 33
1 2 3 4 5 15
duong@ubuntu:~/Desktop/test1$ ./p1
```

```
duong@ubuntu: ~/Desktop/test1
3 4 5 6 7 8 33
3 4 5 6 7 8 33
duong@ubuntu:~/Desktop/test1$ ./p0
3 4 5 12
1 2 3 4 5 15
1 6 7 9 23
3 4 5 6 7 8 33
duong@ubuntu:~/Desktop/test1$ cat file.out
3 4 5 12
1 2 3 4 5 15
1 6 7 9 23
3 4 5 6 7 8 33
duong@ubuntu:~/Desktop/test1$
```

```

1  #include<stdio.h>
2  #include<stdlib.h>
3  #include<string.h>
4  #include<sys/ipc.h>
5  #include<sys/msg.h>
6  #include<sys/errno.h>
7  #define maxx 100
8  struct message_buffer{
9      long mesg_type;
10     int data[maxx];
11     int count;
12 }message;
13 int main(){
14     int a[maxx][maxx];
15     int m=0,n=0;
16     key_t key1=ftok(".",1);
17     int msgid1=msgget(key1,0666 | IPC_CREAT);
18     FILE*f=fopen("file.inp","rt");
19     int i,j;
20     char line[1024];
21     while (fgets(line, sizeof(line), f) != NULL) {
22
23         message.count = 0;
24         char *token = strtok(line, " \t\n");
25         while (token != NULL) {
26             message.data[message.count] = atoi(token);
27             message.count++;
28             token = strtok(NULL, " \t\n");
29         }
30         message.mesg_type=1;
31         msgsnd(msgid1,&message,sizeof(message),IPC_NOWAIT);
32     }
33     fclose(f);
34     /*-----
35     key_t key2=ftok(".",2);
36     int msgid2=msgget(key2,0666 | IPC_CREAT);
37     int len[maxx]={};
38     int pivot;
39     FILE*g=fopen("file.out","wt");
40     while (1) {
41         int rc=msgrcv(msgid2,&message,sizeof(message),2,IPC_NOWAIT);
42         if(rc==-1 && errno==ENOMSG)
43             break;
44
45         for(i=0;i<message.count;i++)
46             a[m][n++]=message.data[i];
47
48         len[m]=n;//mang luu chieu dai tung dong
49         m++;
50         n=0;
51     }
52     msgctl(msgid2, IPC_RMID, NULL);
53
54     int x,y;
55     for(i=0;i<len[m];i++)
56         for(j=0;j<len[i];j++)
57             printf("%d\t",a[i][j]);
58     }
```

```

54     int x,y;
55     /*sort theo sum tung dong*/
56     for(i=0;i<m-1;i++){
57         for(j=i+1;j<m;j++){
58             if(a[i][len[i]-1]>a[j][len[j]-1]){
59                 if(len[i]>len[j])pivot=len[i];
60                 else pivot=len[j];
61
62                 for(x=0;x<pivot;x++){//swap 2 dong
63                     int tmp1=a[i][x];
64                     a[i][x]=a[j][x];
65                     a[j][x]=tmp1;
66                 }
67                 /*swap len[]*/
68                 int tmp2 = len[i];
69                 len[i] = len[j];
70                 len[j] = tmp2;
71             }
72         }
73     }
74     for(i=0;i<m;i++){//in ra mang 2 chieu
75         for(j=0;j<len[i];j++)
76             printf("%d ", a[i][j]);
77         printf("\n");
78     }
79
80     for(i=0;i<m;i++){
81         for(j=0;j<len[i];j++)
82             fprintf(g,"%d ",a[i][j]);
83         fprintf(g,"\n");
84     }
85     fclose(g);
86     return 0;
87 }

```



```

1 #include <stdio.h>
2 #include <stdlib.h>
3 #include <string.h>
4 #include <sys/ipc.h>
5 #include <sys/msg.h>
6 #include <sys/errno.h>
7 #define maxx 100
8
9 struct message_buffer {
10     long msg_type;
11     int data[maxx];
12     int count;
13 } message;
14
15 int main() {
16     int i,j,sum = 0;
17     key_t key1=ftok(".",1);
18     int msgid1=msgget(key1,0666 | IPC_CREAT);
19
20     key_t key2=ftok(".",2);
21     int msgid2=msgget(key2,0666 | IPC_CREAT);
22
23     while (1) { //chu y IPC_NOWAIT nen msgrcv ko cho msgsnd
24         int rc=msgrcv(msgid1,&message,sizeof(message),1,IPC_NOWAIT);
25         if(rc==-1 && errno==ENOMSG)
26             break;
27
28         for(i=0;i<message.count-1;i++){ //sort
29             for(j=i+1;j<message.count;j++){
30                 if(message.data[i]>message.data[j]){
31                     if(message.data[i]>message.data[j]){
32                         int tmp=message.data[i];
33                         message.data[i]=message.data[j];
34                         message.data[j]=tmp;
35                     }
36                 }
37             }
38             sum+=message.data[i];
39             message.data[message.count++]=sum; //them sum vao
40             for(i=0;i<message.count;i++){
41                 printf("%d ",message.data[i]);
42             }
43             printf("\n");
44             message.msg_type=2;
45             msgsnd(msgid2,&message,sizeof(message),IPC_NOWAIT);
46             sum=0;
47         }
48         msgctl(msgid1, IPC_RMID, NULL);
49         return 0;
50     }
51 }

```