**TRƯỜNG ĐẠI HỌC TÔN ĐỨC THẮNG**

**KHOA CÔNG NGHỆ THÔNG TIN**



**BÀI TẬP MÔN HỌC NMHĐH**

**LAB 7**

**Nhóm:** **22** **Tổ: 01**

**Tên : Nguyễn Thị Anh Thư**

**MSSV: 51900564**

**NỘI DUNG BÁO CÁO KẾT QUẢ**

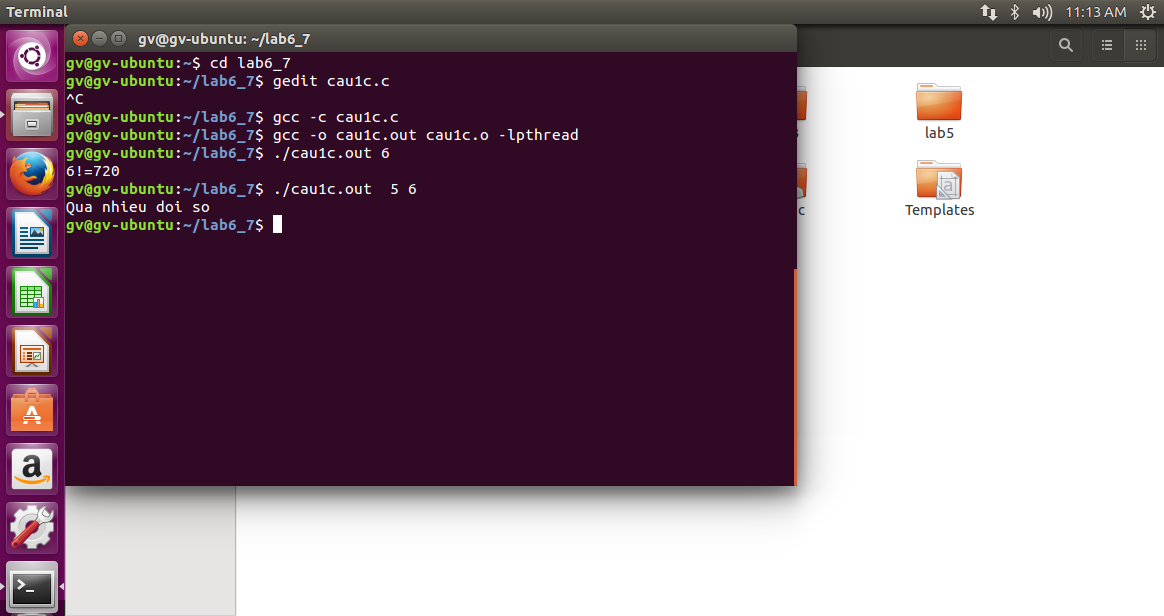
**Bài 1C. Giải lại vấn đề 1A với kỹ thuật truyền thông điệp (Message Passing)**

**KẾT QUẢ THỰC HIỆN**

**1c.1. Phần source code:**

1. #include <stdio.h>
2. #include <sys/ipc.h>
3. #include <sys/msg.h>
4. #include <sys/types.h>
5. #include <sys/wait.h>
6. #include <stdlib.h>
7. #include <unistd.h>
8. struct mesg\_buffer
9. {
10. long mesg\_type;
11. int mesg;
12. };
13. int main(int argc, char \*\* argv)
14. {
15. struct mesg\_buffer mess;
16. key\_t key;
17. int gt = 1;
18. int msgid;
19. key = ftok(".", 1);
20. msgid= msgget(key, 0666| IPC\_CREAT);
21. mess.mesg\_type = 1;
22. int n= atoi(argv[1]);
23. if(argc > 2){
24. printf("Qua nhieu doi so \n");
25. }
26. else if(argc <2){
27. printf("thieu doi so\n");
28. }else{
29. if(n <= 3){
30. printf("Doi so nho hon 3\n");
31. }else{
32. pid\_t pid;
33. pid=fork();
34. if(pid < 0)
35. {
36. printf("Fork failed");
37. }
38. else if(pid==0){
39. int i;
40. for (i=1;i<=n;i++) gt=gt\*i;
41. mess.mesg= gt;
42. msgsnd(msgid, &mess, sizeof(mess), 0);
43. }else{
44. wait(NULL);
45. msgrcv(msgid, &mess, sizeof(mess), 1, 0);
46. printf("%d!=%d\n",n,mess.mesg);
47. msgctl(msgid, IPC\_RMID, NULL);
48. }
49. }
50. }
51. return 0;
52. }

**1c.2. Kết quả chạy chương trình:**

******

**Bài 2c**

2A. Tiến trình cha đọc hai số nguyên và một thao tác +, -, \*, / và chuyển tất cả cho tiến trình con. Quá trình

con tính toán kết quả và trả về cho tiến trình cha. Quá trình cha mẹ ghi kết quả vào một tệp.

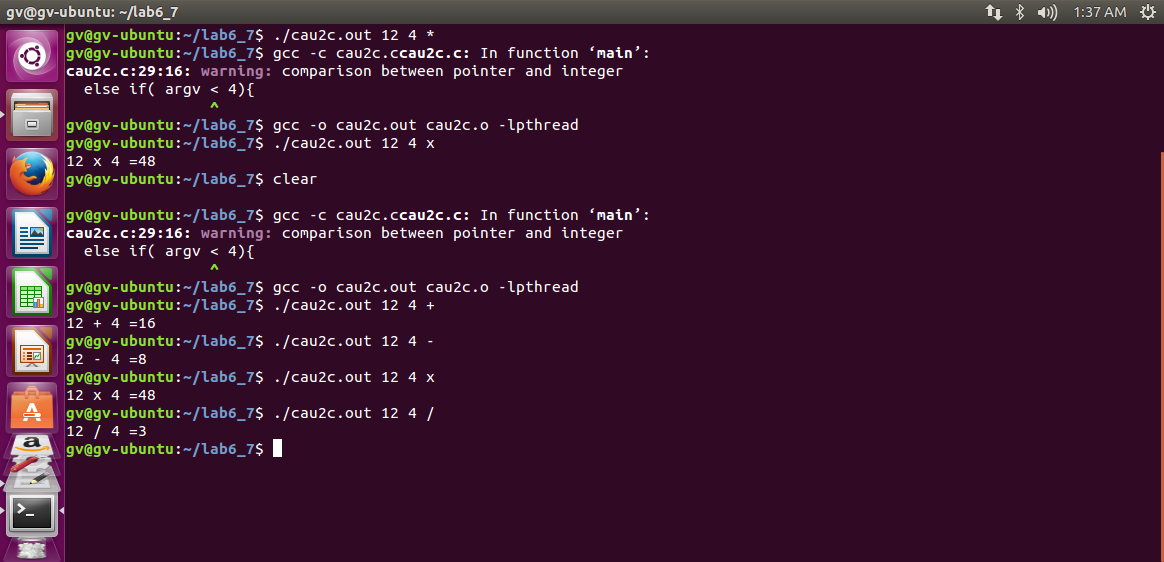
2C.Giải lại vấn đề 1A với kỹ thuật truyền thông điệp (Message Passing)

**KẾT QUẢ THỰC HIỆN**

**2c.1 Phần source code:**

1. #include <stdio.h>
2. #include <sys/ipc.h>
3. #include <sys/msg.h>
4. #include <sys/types.h>
5. #include <sys/wait.h>
6. #include <stdlib.h>
7. #include <unistd.h>
8. #include <string.h>
9. struct mesg\_buffer
10. {
11. long mesg\_type;
12. int mesg\_a,mesg\_b;
13. int mesg\_kq;
14. char mesg[1];
15. };
16. int main(int argc, char \*\* argv)
17. {
18. struct mesg\_buffer mess;
19. key\_t key;
20. int gt = 1;
21. int msgid;
22. key = ftok(".", 1);
23. msgid= msgget(key, 0666| IPC\_CREAT);
24. mess.mesg\_type = 1;
25. int n= atoi(argv[1]);
26. if(argc > 4){
27. printf(" Qua nhieu doi so");
28. }
29. else if( argv < 4){
30. printf(" Thieu doi so");
31. }
32. else{
33. pid\_t pid;
34. pid= fork();
35. if(pid <0){
36. printf(" Fork failed");
37. }
38. else if(pid ==0){
39. sleep(2);
40. msgrcv(msgid,&mess, sizeof(mess),1,0);
41. int a=mess.mesg\_a;
42. int b=mess.mesg\_b;
43. if(strcmp(mess.mesg,"+")==0) mess.mesg\_kq = a+b;
44. else if(strcmp(mess.mesg,"-")==0) mess.mesg\_kq = a-b;
45. else if(strcmp(mess.mesg,"x")==0) mess.mesg\_kq = a\*b;
46. else
47. mess.mesg\_kq= a/b;
48. mess.mesg\_type = 2;
49. msgsnd(msgid, &mess, sizeof(mess),0);
50. }
51. else{
52. mess.mesg\_type =1;
53. mess.mesg\_a= atoi(argv[1]);
54. mess.mesg\_b= atoi(argv[2]);
55. strcpy(mess.mesg,argv[3]);
56. msgsnd(msgid,&mess, sizeof(mess),0);
57. sleep(2);
58. msgrcv(msgid,&mess, sizeof(mess),2,0);
59. printf("%d %s %d =%d\n",mess.mesg\_a, mess.mesg, mess.mesg\_b, mess.mesg\_kq);
60. msgctl(msgid, IPC\_RMID, NULL );
61. }
62. }
63. return 0;
64. }

**2c.2 Kết quả chạy chương trình:**

****

**Bài 3**: (Shared Memory) Tiến trình con ghi một mảng vào SM\_0, với SM[0] chứa số phần tử mảng. Tiến trình cha

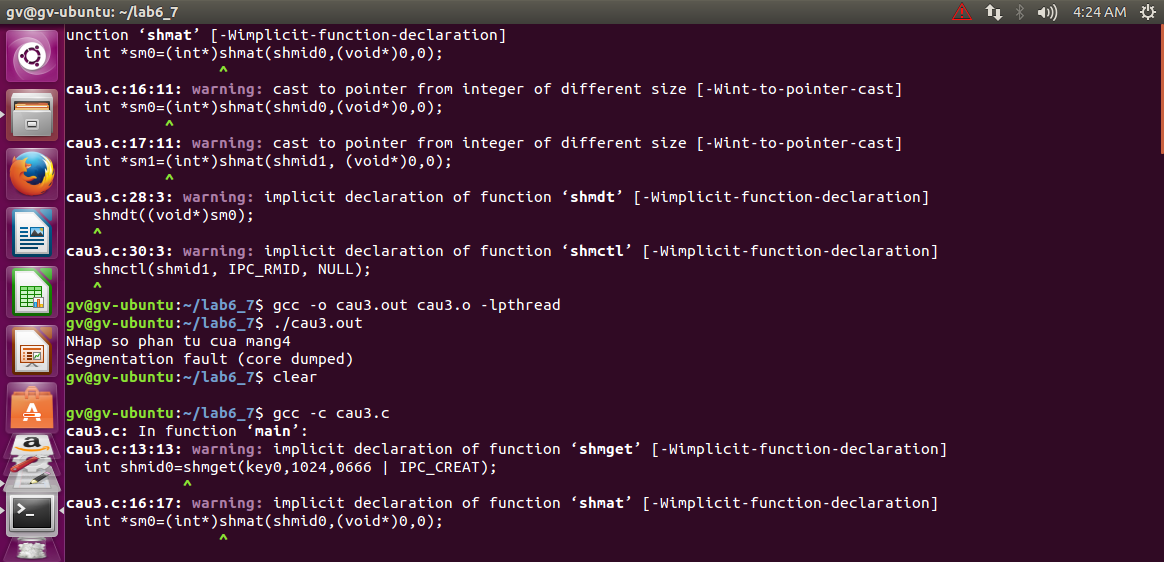
thực hiện, tính tổng các phần tử của mảng và ghi vào cuối SM\_1. Trình con trình nhận và xuất dữ liệu ra màn

hình (lưu ý sử dụng 2 SM).

**KẾT QUẢ THỰC HIỆN**

* 1. **Phần source code:**

1. #include <stdio.h>
2. #include <sys/ipc.h>
3. #include <sys/msg.h>
4. #include <sys/types.h>
5. #include <sys/wait.h>
6. #include <stdlib.h>
7. #include <unistd.h>
8. #include <string.h>
9. int main(){
10. key\_t key0 = ftok("shmfile0", 67);
11. key\_t key1 = ftok("shmfile0", 65);
12. int shmid0= shmget(key0, 1024, 0666 | IPC\_CREAT);
13. int shmid1= shmget(key1, 1024, 0666 | IPC\_CREAT);
14. int \*sm0=(int\*)shmat(shmid0,(void\*)0,0);
15. int \*sm1=(int\*)shmat(shmid1, (void\*)0,0);
16. int pid;
17. pid=fork();
18. if(pid<0){
19. printf("Fork failed\n ");
20. return -1;
21. }
22. else if(pid ==0){
23. printf("NHap so phan tu cua mang : ");
24. scanf("%d", &sm0[0]);
25. for(int i = 1; i <= sm0[0]; i++){
26. printf("a[%d]: = ", i);
27. scanf("%d", &sm0[i]);
28. }
29. sleep(8);
30. printf("Tong cac phan tu = %d\n", sm1[0]);
31. shmdt((void\*)sm0);
32. shmdt((void\*)sm1);
33. shmctl(shmid1, IPC\_RMID, NULL);
34. }
35. else {
36. sleep(10);
37. int sum= 0;
38. for( int i=1;i<=sm0[0]; i++){
39. sum=sum+sm0[i];
40. }
41. sm1[0]=sum;
42. shmdt((void\*)sm0);
43. shmdt((void\*)sm1);
44. shmctl(shmid0, IPC\_RMID, NULL);
45. wait(NULL);
46. }
47. return 0;
48. }
    1. **Kết quả chạy chương trình:**



**KẾT LUẬN**

Qua lab6\_7 em hiểu được cách sử dụng unnamed pipe và named pipe, cách chia sẻ vùng nhớ và hàng đợi message.