Gebze Technical University
Computer Engineering

CSE 344 - 2021 Spring

HOMEWORK 3 REPORT

BERKE SÜSLÜ
161044076

1 INTRODUCTION

1.1 Problem Definition

Hot potato game using shared memory and named semaphore.

1.2 System Requirements

Any computer with Ubuntu 14.04 LTS 32-bit Operating System.

2 METHOD

2.1 Problem Solution Approach

Firstly,my program reads the all parameters using getopt(). If the parameters are missing or invalid, the program exits with error. The program opens the named semaphore(creates if it is not created).
My program reads the fifo list from the path(the path is in the -f parameter). If the shared memory is not created, the program creates once and fill it with fifo paths and select a fifo path. If the shared memory is already created, the program reads shared memory and select an unused fifo path.
If the process has a potato,it will be initialized in shared memory(potato id and cooldown switches).
Then,process with a potato opens a random fifo(not himself) and send the potato id into fifo.(using write() function)
Process without a potato is wait for a potato. If the potato has come into his fifo, it reads the fifo and decrements the cooldown switches of the potato.
When all potatoes are done, the last process sends a message to other process (I choose “-1” for this) and frees all resources.
If there are more than 1 potato, sometimes program can be deadlock but program works well with 1 potato.

3 RESULT

3.1 Test Cases

I used an example fifolist file and checked with valgrind for memory leaks.

3.2 Running Results

The program executes:

```
cse312@ubuntu:~/Desktop/HW3$ ./player -b 12 -s t11 -f fifolist -m t11 > debug &
./player -b 0 -s t11 -f fifolist -m t11 > debug2 & ./player -b 0 -s t11 -f fifolist -m t11 > debug3
```

Terminal outputs(3 different process):

debug

```
pid=4730 current fifo= secondinsecondout
pid=4730 sending potato number 4730 to firstinfirstout; this is switch number 1
pid=4730 receiving potato number 4730 from secondinsecondout
pid=4730 sending potato number 4730 to firstinfirstout; this is switch number 3
pid=4730 receiving potato number 4730 from secondinsecondout
pid=4730 sending potato number 4730 to thirdinthirdout; this is switch number 5
pid=4730 receiving potato number 4730 from secondinsecondout
pid=4730 sending potato number 4730 to thirdinthirdout; this is switch number 12
No left potatoes, exiting.
```

debug3

```
pid=4732 current fifo= firstinfirstout
pid=4732 receiving potato number 4730 from firstinfirstout
pid=4732 sending potato number 4730 to secondinsecondout; this is switch number 2
pid=4732 receiving potato number 4730 from firstinfirstout
pid=4732 sending potato number 4730 to secondinsecondout; this is switch number 4
pid=4732 receiving potato number 4730 from firstinfirstout
pid=4732 sending potato number 4730 to thirdinthirdout; this is switch number 7
pid=4732 receiving potato number 4730 from firstinfirstout
pid=4732 sending potato number 4730 to thirdinthirdout; this is switch number 9
pid=4732 receiving potato number 4730 from firstinfirstout
pid=4732 sending potato number 4730 to secondinsecondout; this is switch number 11
No left potatoes, exiting.
```

debug2

```
pid=4731 current fifo= thirdinthirdout
pid=4731 receiving potato number 4730 from thirdinthirdout
pid=4731 sending potato number 4730 to firstinfirstout; this is switch number 6
pid=4731 receiving potato number 4730 from thirdinthirdout
pid=4731 sending potato number 4730 to firstinfirstout; this is switch number 8
pid=4731 receiving potato number 4730 from thirdinthirdout
pid=4731 sending potato number 4730 to firstinfirstout; this is switch number 10
pid=4731 receiving potato number 4730 from thirdinthirdout
pid=4731; potato number 4730 has cooled down.
All potatoes are cold.
Freeing all resources.
```