General Strategy Psoudecode

Assumptions:

- filewithfifonames contains exact same number of fifonames with processes
- At most 1024 character in each line of fileWithFifoNames
- At most 200 processes are supported

Some Constants that I used

```
#define NAME_MAX_LEN 255
#define LINE_MAX_LEN 1024
#define MAX_NUM_OF_PROCESS 200
```

Struct declerations

```
struct fifoInfo{
        char name[NAME_MAX_LEN];
        Int ownerPid;
        int readEndLastOpened; // if the value is 1, first open WR end of the other fifos then open RD end of this fifo
};

struct potatoInfo{
        int id;
        int temperature; // current temperature
        int doneSwitchNumber;
};

struct message{
        int senderPid;
        int potatoId;
};

struct sharedMem{
        sem_t semFifoBarrier;
        int numOfProceses;
        struct fifoInfo fifos[MAX_NUM_OF_PROCESS];
        struct potatoInfo potatos[MAX_NUM_OF_PROCESS];
};
```

Each process creates its Fifo file,

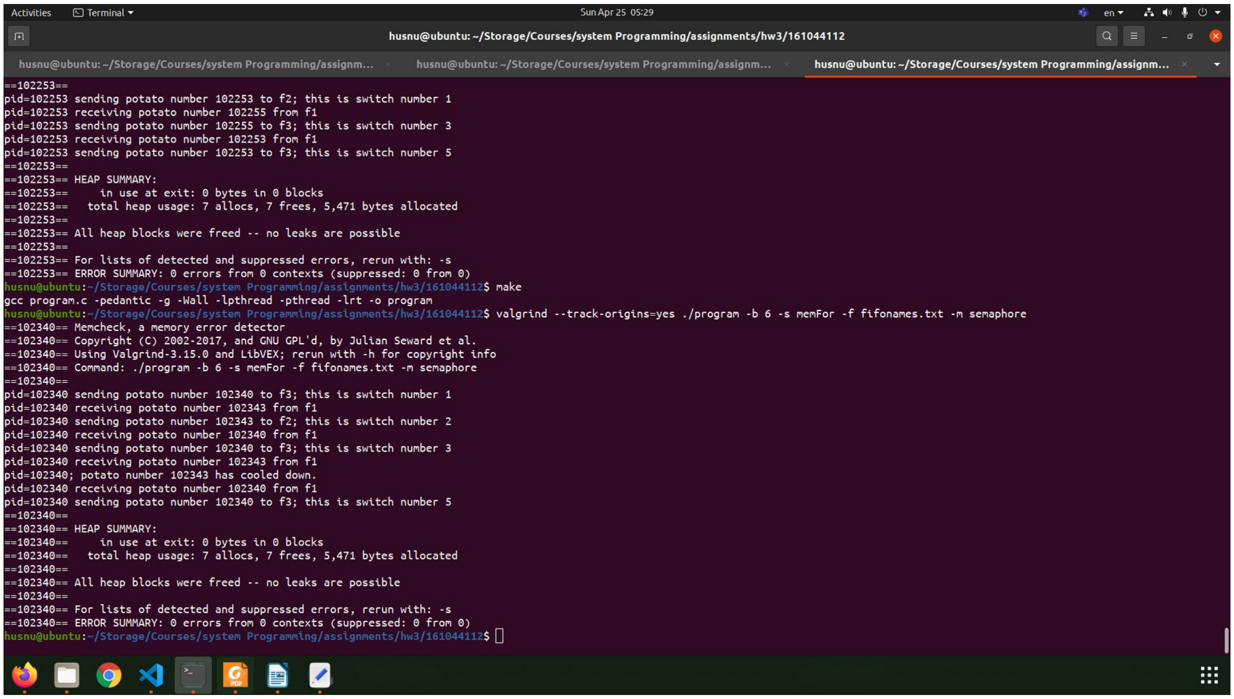
First executed processes is responsible for waiting for others to create their fifos, then it opens all other fifoes write ends. Then open its read end. On the other hand all the other processes opens their read ends first then open all the fifos write ends.

For waiting fifo creation of other processes an unnamed semaphore is used. The process who needs to wait perform sem_wait() n-1 times, and the other n-1 processes perform sem_post() on that unnamed semaphore one by one.

Temperature of potatos and number of switch that is made so far is maintained in shared memory segment.

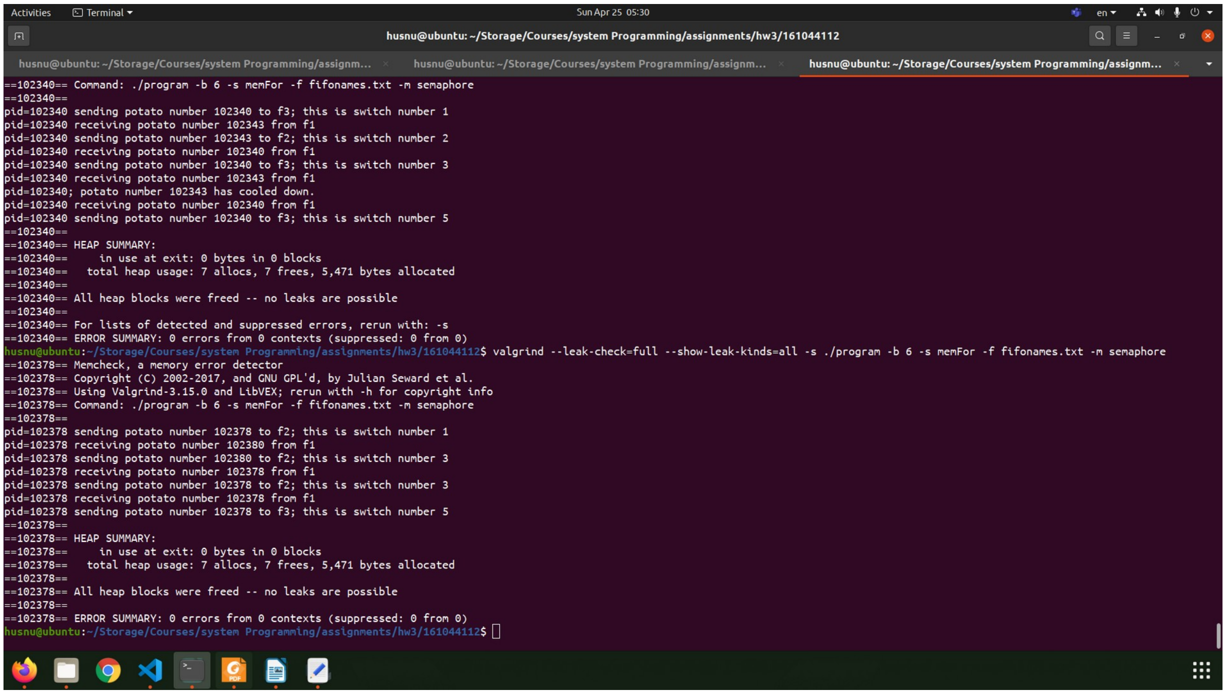
While creating shared memory segment, reading file, flock() is used for synchronization.

Screan Shots



Activities
Terminal
Sun Apr 25 05:29
en
husnu@ubuntu: ~/Storage/Courses/system Programming/assignments/hw3/161044112
husnu@ubuntu: ~/Storage/Courses/system Programming/assignm...
husnu@ubuntu: ~/Storage/Courses/system Programming/assignm...
husnu@ubuntu: ~/Storage/Courses/system Programming/assignm...

```
==102253==
pid=102253 sending potato number 102253 to f2; this is switch number 1
pid=102253 receiving potato number 102255 from f1
pid=102253 sending potato number 102255 to f3; this is switch number 3
pid=102253 receiving potato number 102253 from f1
pid=102253 sending potato number 102253 to f3; this is switch number 5
==102253==
==102253== HEAP SUMMARY:
==102253==     in use at exit: 0 bytes in 0 blocks
==102253==   total heap usage: 7 allocs, 7 frees, 5,471 bytes allocated
==102253==
==102253== All heap blocks were freed -- no leaks are possible
==102253==
==102253== For lists of detected and suppressed errors, rerun with: -s
==102253== ERROR SUMMARY: 0 errors from 0 contexts (suppressed: 0 from 0)
husnu@ubuntu:~/Storage/Courses/system Programming/assignments/hw3/161044112$ make
gcc program.c -pedantic -g -Wall -lpthread -pthread -lrt -o program
husnu@ubuntu:~/Storage/Courses/system Programming/assignments/hw3/161044112$ valgrind --track-origins=yes ./program -b 6 -s memFor -f fifonames.txt -m semaphore
==102340== Memcheck, a memory error detector
==102340== Copyright (C) 2002-2017, and GNU GPL'd, by Julian Seward et al.
==102340== Using Valgrind-3.15.0 and LibVEX; rerun with -h for copyright info
==102340== Command: ./program -b 6 -s memFor -f fifonames.txt -m semaphore
==102340==
pid=102340 sending potato number 102340 to f3; this is switch number 1
pid=102340 receiving potato number 102343 from f1
pid=102340 sending potato number 102343 to f2; this is switch number 2
pid=102340 receiving potato number 102340 from f1
pid=102340 sending potato number 102340 to f3; this is switch number 3
pid=102340 receiving potato number 102343 from f1
pid=102340; potato number 102343 has cooled down.
pid=102340 receiving potato number 102340 from f1
pid=102340 sending potato number 102340 to f3; this is switch number 5
==102340==
==102340== HEAP SUMMARY:
==102340==     in use at exit: 0 bytes in 0 blocks
==102340==   total heap usage: 7 allocs, 7 frees, 5,471 bytes allocated
==102340==
==102340== All heap blocks were freed -- no leaks are possible
==102340==
==102340== For lists of detected and suppressed errors, rerun with: -s
==102340== ERROR SUMMARY: 0 errors from 0 contexts (suppressed: 0 from 0)
husnu@ubuntu:~/Storage/Courses/system Programming/assignments/hw3/161044112$
```


Activities
Terminal
Sun Apr 25 05:30
en
husnu@ubuntu: ~/Storage/Courses/system Programming/assignments/hw3/161044112
husnu@ubuntu: ~/Storage/Courses/system Programming/assignm...
husnu@ubuntu: ~/Storage/Courses/system Programming/assignm...
husnu@ubuntu: ~/Storage/Courses/system Programming/assignm...

```
==102340== Command: ./program -b 6 -s memFor -f fifonames.txt -m semaphore
==102340==
pid=102340 sending potato number 102340 to f3; this is switch number 1
pid=102340 receiving potato number 102343 from f1
pid=102340 sending potato number 102343 to f2; this is switch number 2
pid=102340 receiving potato number 102340 from f1
pid=102340 sending potato number 102340 to f3; this is switch number 3
pid=102340 receiving potato number 102343 from f1
pid=102340; potato number 102343 has cooled down.
pid=102340 receiving potato number 102340 from f1
pid=102340 sending potato number 102340 to f3; this is switch number 5
==102340==
==102340== HEAP SUMMARY:
==102340==     in use at exit: 0 bytes in 0 blocks
==102340==   total heap usage: 7 allocs, 7 frees, 5,471 bytes allocated
==102340==
==102340== All heap blocks were freed -- no leaks are possible
==102340==
==102340== For lists of detected and suppressed errors, rerun with: -s
==102340== ERROR SUMMARY: 0 errors from 0 contexts (suppressed: 0 from 0)
husnu@ubuntu:~/Storage/Courses/system Programming/assignments/hw3/161044112$ valgrind --leak-check=full --show-leak-kinds=all -s ./program -b 6 -s memFor -f fifonames.txt -m semaphore
==102378== Memcheck, a memory error detector
==102378== Copyright (C) 2002-2017, and GNU GPL'd, by Julian Seward et al.
==102378== Using Valgrind-3.15.0 and LibVEX; rerun with -h for copyright info
==102378== Command: ./program -b 6 -s memFor -f fifonames.txt -m semaphore
==102378==
pid=102378 sending potato number 102378 to f2; this is switch number 1
pid=102378 receiving potato number 102380 from f1
pid=102378 sending potato number 102380 to f2; this is switch number 3
pid=102378 receiving potato number 102378 from f1
pid=102378 sending potato number 102378 to f2; this is switch number 3
pid=102378 receiving potato number 102378 from f1
pid=102378 sending potato number 102378 to f3; this is switch number 5
==102378==
==102378== HEAP SUMMARY:
==102378==     in use at exit: 0 bytes in 0 blocks
==102378==   total heap usage: 7 allocs, 7 frees, 5,471 bytes allocated
==102378==
==102378== All heap blocks were freed -- no leaks are possible
==102378==
==102378== ERROR SUMMARY: 0 errors from 0 contexts (suppressed: 0 from 0)
husnu@ubuntu:~/Storage/Courses/system Programming/assignments/hw3/161044112$ 
```