207SE – OPERATING SYSTEMS,SECURITY AND NETWORKS

Portfolio 2

Abstract

A second Course Work Portfolio for 207SE Module, Computer Science 2nd year

Rokas Navickas

Student ID: 7868629

Rokas Navickas
207SE – Operating Systems, Security and Networks
Portfolio 2
Student ID: 7868629

LAB 13 – Fork Menu Systems:

Basic Task:

Ls,who,date

The evidence of program running(full code will be in advanced section):

```
navickar@hvs-its-lnx01:~/207SE_Sessions/Session13/fork_code$ ./fork
1. ls
2. who
3. date
4. ping
5. dual ping
6. killing spree
1
 I am the parent my PID is 11284, myPPID is 62723,
 Mychild process has finished.
  I am the child , my PID is 11309 , my PPID is 11284
navickar@hvs-its-lnx01:~/207SE_Sessions/Session13/fork_code$ for_fork.c  fork  fork_execl.c    simple_execl.c  twoproc.c  twoproc_pid.c
```

```
navickar@hvs-its-lnx01:~/207SE_Sessions/Session13/fork_code$ ./fork
1. ls
2. who
3. date
4. ping
5. dual ping
6. killing spree
2
 I am the parent my PID is 11402, myPPID is 62723,
 Mychild process has finished.
  I am the child , my PID is 11409 , my PPID is 11402
navickar@hvs-its-lnx01:~/207SE_Sessions/Session13/fork_code$ saidih3  pts/0        2019-03-21 21:20 (10.4.12.186)
narvilak pts/2        2019-03-21 22:18 (10.4.13.19)
navickar pts/3        2019-03-21 19:50 (10.4.13.70)
jagneb   pts/4        2019-03-21 19:07 (10.4.13.82)
limacarg pts/5        2019-03-21 22:19 (10.0.34.74)
stewar83 pts/6        2019-03-21 22:22 (10.4.13.51)
reeses   pts/7        2019-03-21 19:26 (10.4.13.30)
machina2 pts/8        2019-03-21 21:25 (10.4.13.60)
machina2 pts/10       2019-03-21 21:25 (10.4.13.60)
sayam    pts/12       2019-03-21 20:46 (10.0.61.78)
stewar83 pts/15       2019-03-21 22:22 (10.4.13.51)
stewar83 pts/18       2019-03-21 22:22 (10.4.13.51)
```

```
navickar@hvs-its-lnx01:~/207SE_Sessions/Session13/fork_code$ ^C
navickar@hvs-its-lnx01:~/207SE_Sessions/Session13/fork_code$ ./fork
1. ls
2. who
3. date
4. ping
5. dual ping
6. killing spree
3
 I am the parent my PID is 11499, myPPID is 62723,
 Mychild process has finished.
  I am the child , my PID is 11500 , my PPID is 11499
navickar@hvs-its-lnx01:~/207SE_Sessions/Session13/fork_code$ Thu 21 Mar 23:02:12 GMT 2019
```

Ping

Ping evidence:

```
navickar@hvs-its-lnx01:~/207SE_Sessions/Session13/fork_code$ ^C
navickar@hvs-its-lnx01:~/207SE_Sessions/Session13/fork_code$ ./fork
1. ls
2. who
3. date
4. ping
5. dual ping
6. killing spree
4
 I am the parent my PID is 11579, myPPID is 62723,
 Mychild process has finished.
  I am the child , my PID is 11592 , my PPID is 11579
navickar@hvs-its-lnx01:~/207SE_Sessions/Session13/fork_code$ PING d2bytcopxu066p.cloudfront.net (54.192.8.250) 56(84) bytes of data.
64 bytes from server-54-192-8-250.lhr3.r.cloudfront.net (54.192.8.250): icmp_seq=1 ttl=245 time=6.43 ms
64 bytes from server-54-192-8-250.lhr3.r.cloudfront.net (54.192.8.250): icmp_seq=2 ttl=245 time=6.67 ms
64 bytes from server-54-192-8-250.lhr3.r.cloudfront.net (54.192.8.250): icmp_seq=3 ttl=245 time=6.49 ms
64 bytes from server-54-192-8-250.lhr3.r.cloudfront.net (54.192.8.250): icmp_seq=4 ttl=245 time=6.45 ms
64 bytes from server-54-192-8-250.lhr3.r.cloudfront.net (54.192.8.250): icmp_seq=5 ttl=245 time=6.57 ms
64 bytes from server-54-192-8-250.lhr3.r.cloudfront.net (54.192.8.250): icmp_seq=6 ttl=245 time=6.58 ms
64 bytes from server-54-192-8-250.lhr3.r.cloudfront.net (54.192.8.250): icmp_seq=7 ttl=245 time=6.78 ms
64 bytes from server-54-192-8-250.lhr3.r.cloudfront.net (54.192.8.250): icmp_seq=8 ttl=245 time=6.53 ms
64 bytes from server-54-192-8-250.lhr3.r.cloudfront.net (54.192.8.250): icmp_seq=9 ttl=245 time=6.84 ms
64 bytes from server-54-192-8-250.lhr3.r.cloudfront.net (54.192.8.250): icmp_seq=10 ttl=245 time=7.36 ms
64 bytes from server-54-192-8-250.lhr3.r.cloudfront.net (54.192.8.250): icmp_seq=11 ttl=245 time=6.49 ms
64 bytes from server-54-192-8-250.lhr3.r.cloudfront.net (54.192.8.250): icmp_seq=12 ttl=245 time=6.46 ms
```

Rokas Navickas
207SE – Operating Systems, Security and Networks
Portfolio 2
Student ID: 7868629

Advanced Task:

Code for whole program:

```
//fork code that uses multiple loops to creates parents and children
#include <unistd.h>
#include <stdio.h>
#include <string.h>

int main(){
    char option[1];//Storing user option here
    printf("1. ls\n");
    printf("2. who\n");
    printf("3. date\n");
    printf("4. ping\n");
    printf("5. dual ping\n");
    printf("6. killing spree\n");
    scanf("%s",option);
    if (strcmp(option,"1")==0){
        int pid=fork();
        if(pid!=0) {//part of  fork_execl.c file
            printf ( " I am the parent my PID is %d, myPPID is %d, \n
",getpid(),getppid());//PID and Parent PID:
            printf( "Mychild process has finished. \n ");
        }
        else {
            printf ( " I am the child , my PID is %d , my PPID is %d
\n",getpid(),getppid());//Child and Child Parent PID:
            sleep(2);
            execl ( "/bin/ls",".",(char*)0);//Runs ls using execl
        }
    }
    else if(strcmp(option,"2")==0){
        int pid=fork();
        if(pid!=0) {//part of  fork_execl.c file
            printf ( " I am the parent my PID is %d, myPPID is %d, \n
",getpid(),getppid());//PID and Parent PID:
            printf( "Mychild process has finished. \n ");
        }else {
            printf ( " I am the child , my PID is %d , my PPID is %d
\n",getpid(),getppid());//Child and Child Parent PID:
            execl ( "/usr/bin/who",".",(char*)0);//Runs who -u using execl

        }

    }
    else if(strcmp(option,"3")==0){
        int pid=fork();
        if(pid!=0) {//part of  fork_execl.c file
```

```
            printf ( " I am the parent my PID is %d, myPPID is %d, \n
",getpid(),getppid());//PID and Parent PID:
            printf( "Mychild process has finished. \n ");
        }
        else {
            printf ( " I am the child , my PID is %d , my PPID is %d
\n",getpid(),getppid());//Child and Child Parent PID:
            execl ( "/bin/date","date",0,(char*)NULL);//Runs top using execl
        }
    }
    else if(strcmp(option,"4")==0){
        int pid=fork();
        if(pid!=0) {//part of  fork_execl.c file
            printf ( " I am the parent my PID is %d, myPPID is %d, \n
",getpid(),getppid());//PID and Parent PID:
            printf( "Mychild process has finished. \n ");
        }
        else {
            printf ( " I am the child , my PID is %d , my PPID is %d
\n",getpid(),getppid());//Child and Child Parent PID:
            sleep(2);
            execl ( "/bin/ping","ping","www.imdb.com",(char*)0);//Pings imdb
using execl
        }
    }
    else if(strcmp(option,"5")==0){
        int pid=fork();
        if(pid!=0) {//part of  fork_execl.c file
            printf ( " I am the parent my PID is %d, myPPID is %d, \n
",getpid(),getppid());//PID and Parent PID:
            execl ( "/bin/ping","ping","uuu.shef.ac.uk",(char*)0);//Pings
sheffield.ac.uk
        }
        else {
            printf ( " I am the child , my PID is %d , my PPID is %d
\n",getpid(),getppid());//Child and Child Parent PID:
            execl ( "/bin/ping","ping","www.shu.ac.uk",(char*)0);//Pings
shu.ac.uk
        }
    }
    else if(strcmp(option,"6")==0){
       int pid=fork();
        if(pid!=0) {//part of  fork_execl.c file
            printf ( " I am the parent my PID is %d, myPPID is %d, \n
",getpid(),getppid());//PID and Parent PID:
        }
        else {
```

```
            printf ( " I am the child , my PID is %d , my PPID is %d
\n",getpid(),getppid());//Child and Child Parent PID:
            execl ( "/bin/kill","kill",getpid(),(char*)0);
            printf ( " I am the child , my PID is %d , my PPID is %d
\n",getpid(),getppid());//Child and Child Parent PID:
        }
    }
    else{


    }
    return 0;
}
```

Given hostname of sheffield was invalid, hence I looked after a working one:

Results for checks on www.sheffield.ac.uk

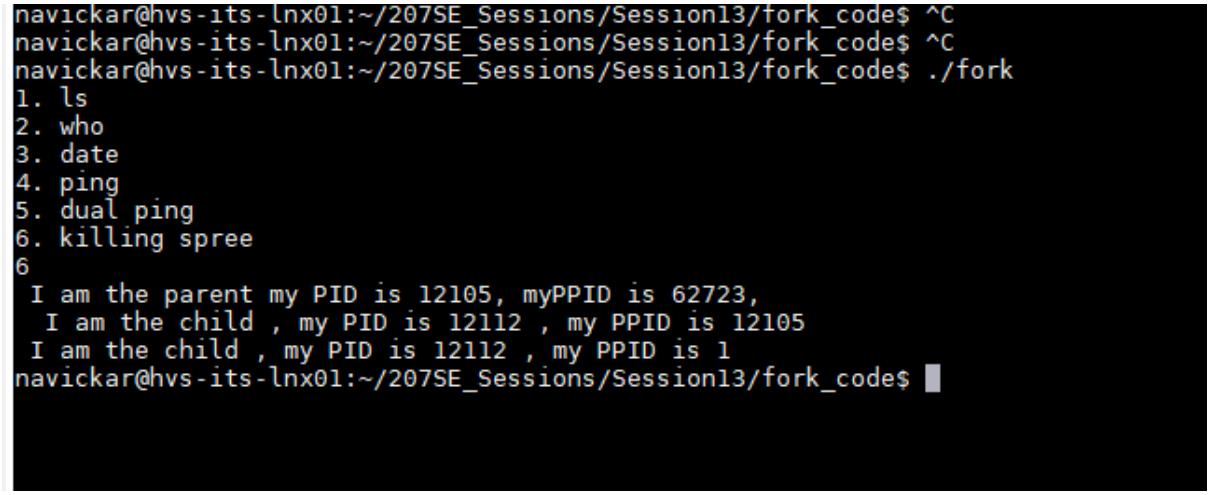
Host	TTL	Class	Type	Details
uuu.shef.ac.uk	600	IN	A	143.167.2.102

Pinged to shef and shu from two different processed:

```
navickar@hvs-its-lnx01:~/207SE_Sessions/Session13/fork_code$ ./fork
1. ls
2. who
3. date
4. ping
5. dual ping
6. killing spree
5
 I am the parent my PID is 11969, myPPID is 62723,
 I am the child , my PID is 11976 , my PPID is 11969
PING uuu.shef.ac.uk (143.167.2.102) 56(84) bytes of data.
PING www.shu.ac.uk (168.63.48.113) 56(84) bytes of data.
64 bytes from uuu.shef.ac.uk (143.167.2.102): icmp_seq=1 ttl=54 time=12.2 ms
64 bytes from uuu.shef.ac.uk (143.167.2.102): icmp_seq=2 ttl=54 time=12.2 ms
64 bytes from uuu.shef.ac.uk (143.167.2.102): icmp_seq=3 ttl=54 time=12.2 ms
64 bytes from uuu.shef.ac.uk (143.167.2.102): icmp_seq=4 ttl=54 time=12.3 ms
64 bytes from uuu.shef.ac.uk (143.167.2.102): icmp_seq=5 ttl=54 time=13.1 ms
64 bytes from uuu.shef.ac.uk (143.167.2.102): icmp_seq=6 ttl=54 time=12.6 ms
64 bytes from uuu.shef.ac.uk (143.167.2.102): icmp_seq=7 ttl=54 time=12.2 ms
```

Evidence of parent killing child process working(parent resets to 1, because it was recently killed):

Rokas Navickas
207SE – Operating Systems, Security and Networks
Portfolio 2
Student ID: 7868629


navickar@hvs-its-lnx01:~/207SE_Sessions/Session13/fork_code$ ^C
navickar@hvs-its-lnx01:~/207SE_Sessions/Session13/fork_code$ ^C
navickar@hvs-its-lnx01:~/207SE_Sessions/Session13/fork_code$ ./fork
1. ls
2. who
3. date
4. ping
5. dual ping
6. killing spree
6
I am the parent my PID is 12105, myPPID is 62723,
I am the child , my PID is 12112 , my PPID is 12105
I am the child , my PID is 12112 , my PPID is 1
navickar@hvs-its-lnx01:~/207SE_Sessions/Session13/fork_code$

LAB 14 – Process Manipulation:

Basic Task:

<disown>, <screen> and <nohup> functions – similarities and differences:

Although I have been using <tmux> command instead of these three, they are all have one main reason – to keep the process running even when the user have logged out or have suspended terminal. In this particular scenario, if you have to leave your computer and you have used one of these commands – worry not – your work will not be lost, and your program will continue to run in background even if the sessions has ended.

<disown> is the one that has different working principle. Instead of running the command before starting a process, you run the process, then suspend it with [CTRL] + [Z], send it to background with <bg> command and detach it from the terminal with disown command to leave it running even after terminal is closed.

<screen> command however, has to be opened before-hand. It opens a new "Screen" Window and allows you to run programs in that window in order to detach from the session and keep the program running, this can be done by typing <Ctrl+a d>. This will work essentially the same as nohup.

"Although "*nohup*", sound like "no hope", it means 'no hang-up' so can logout leaving program running." (Bridgett, 2013) To start the process, we need to write <nohup> "nameOfCommand" and we can drop & at the end of the line to drop it straight to the foreground. After we are finished we can pull the program back using <fg> command.

To wrap everything – all three of these commands are used to run files even if after terminal session is closed and to avoid time-outs.

Watch command:

Watch description from the official man page:

"watch runs *command* repeatedly, displaying its output (the first screenfull). This allows you to watch the program output change over time. By default, the program is run every 2 seconds;" (Rems, 1991)

<watch date> will display date every time interval (by default - 2 seconds)

<watch -d> however will highlight the changes in directory. Such as:

<watch -d ls -l> will show contents of a directory every 2 seconds (by default) and will highlight changes if any. If a group of people are working on the same project or directory, this might come in handy to see what and when is being changed.

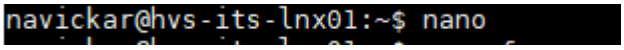
To change refresh interval, we can do it by using -n [number of seconds]. E.g. <watch -n 3600 -d ls -l> this will show changes every hour and highlight changes in current directory.

Watch command might also come in handy with a use of <df>. This would monitor a disk space and can be used if you consider of making a command more efficient. You run a command and observe changes in disk usage of your program. However, this would not give big of an impact on relatively small programs.

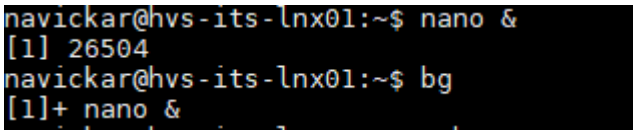
Advanced Task:

Start :

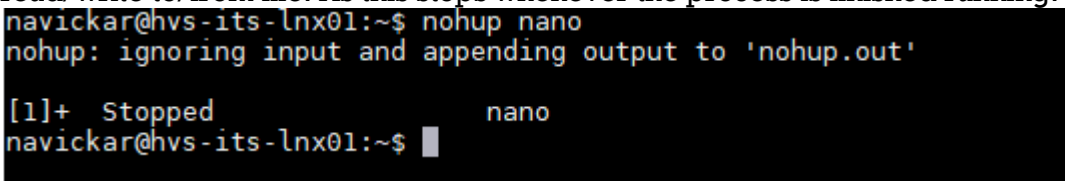
1.The easiest way to start a process is to simply type a name of a process into the CLI:



2. Another way of starting a process is to send it straight to a background and then retrieve the process by writing bg to check if it's running:



3. One more different way to start it while sending it to background is to use nohup command, however, this is only useful to the processes that are continuous such as input read/write to/from file. As this stops whenever the process is finished running.



Suspend:

The easiest way to suspend a program is to press [CTRL+C] while program is running:

```
navickar@hvs-its-lnx01:~$ Thu 21 Mar 00:26:47 GMT 2019
^C
[1]+  Done                    date
```

Another way is to use kill –STOP %(number of process) that can be checked with command of bg.:

```
navickar@hvs-its-lnx01:~$ kill -STOP %1

[1]+  Stopped                 nano
navickar@hvs-its-lnx01:~$
```

And the third way is to use pkill –STOP (name of the process). This is usefull if you do not want to end up suspending the wrong program.:

```
navickar@hvs-its-lnx01:~$ Thu 21 Mar 00:27:20 GMT 2019
pkill -STOP date
[1]+  Done                    date
navickar@hvs-its-lnx01:~$
```

Foreground:

The most simple way to bring a program to foreground is to use & symbol, however, this approach also starts the process in foreground, hence it was used as a start example. Another way of doing it, is to do fg (process id) or fg(name of the process):

```
navickar@hvs-its-lnx01:~$ fg nano
nano
Use "fg" to return to nano.

[1]+  Stopped                 nano
navickar@hvs-its-lnx01:~$
```

Kill:

One way to kill a process Is to use kill –(number of signal) (number of PID)

1 – hangup, 9 – kill signal, 15- terminate.

```
26714 ?        00:00:00 kworker/4:2
26746 ?        00:00:00 kworker/u128:1
26747 pts/1    00:00:00 nano
26757 pts/2    00:00:00 nano
26758 pts/1    00:00:00 ps
navickar@hvs-its-lnx01:~$ kill -15 26747
navickar@hvs-its-lnx01:~$
```

Another way is to use pkill -(number of signal) (number of PID):

```
26764 ?        00:00:00 kworker/u128:0
26766 pts/2    00:00:00 nano
26767 pts/1    00:00:00 nano
26768 pts/1    00:00:00 nano
26783 pts/1    00:00:00 ps
navickar@hvs-its-lnx01:~$ pkill -15 26766
navickar@hvs-its-lnx01:~$
```

We can also use killall, which kills all of the processes that are the same of the input PID:

```
navickar@hvs-its-lnx01:~$ pkill -15 26766
navickar@hvs-its-lnx01:~$ killall -9 nano
nano(26766): Operation not permitted
[1]   Killed                  nano
[2]-  Killed                  nano
[3]+  Killed                  nano
navickar@hvs-its-lnx01:~$
```

Killing a program as a user is efficient way to kill something without a requirement to log out and log in to another user. We can use killall –u (login) (name of the program):

```
navickar@hvs-its-lnx01:~$ killall -u navickar nano
[2]-  Terminated              nano
navickar@hvs-its-lnx01:~$
```

LAB 15 – Synchronisation – Hole in the Bucket:

Basic Task – Hole in the bucket:

Code for the task:

```
//critical_example2.c
#include <sys/ipc.h>
#include <sys/sem.h>
#include <stdio.h>
#include <stdlib.h>
#include <unistd.h>


#include "se207_sems.h"

int main(int argc, char argv[]){
int id;
//Use our source file as the "key"
  id=se207_semget("critical_example2.c",1);

  int pid=fork();
  if(pid){
    //P1

    while(1){

      se207_wait(id);
      rsleep();
      printf("\n");
      se207_signal(id);

      se207_wait(id);
      printf("Yes,Liza!\n");
      se207_signal(id);

      se207_wait(id);
      printf("Don't fetch the water..\n");
      printf("\n");
      printf("There's a hole in the bucket, dear Liza, dear Liza,\n");
      printf("There's a hole in the bucket, dear Liza, a hole.\n");
      se207_signal(id);

      se207_wait(id);
      printf("With what shall I fix it, dear Liza, dear Liza?\n");
      printf("With what shall I fix it, dear Liza, with what?\n");
      se207_signal(id);
```

```
    se207_wait(id);
    printf("But the straw is too long, dear Liza, dear Liza,\n");
    printf("The straw is too long, dear Liza, too long.\n");
    se207_signal(id);

    se207_wait(id);
    printf("With what shall I cut it, dear Liza, dear Liza?\n");
    printf("With what shall I cut it, dear Liza, with what?\n");
    se207_signal(id);

    se207_wait(id);
    printf("The axe is too dull, dear Liza, dear Liza,\n");
    printf("The axe is too dull, dear Liza, too dull.\n");
    se207_signal(id);

    se207_wait(id);
    printf("On what shall I sharpen it, dear Liza, dear Liza?\n");
    printf("On what shall I hone it, dear Liza, with what?\n");
    se207_signal(id);

    se207_wait(id);
    printf("But the stone is too dry, dear Liza, dear Liza,\n");
    printf("The stone is too dry, dear Liza, too dry.\n");
    se207_signal(id);

    se207_wait(id);
    printf("With what shall I wet it, dear Liza, dear Liza?\n");
    printf("With what shall I wet it, dear Liza, with what?\n");
    se207_signal(id);

    se207_wait(id);
    printf("In what shall I fetch it, dear Liza, dear Liza?\n");
    printf("In what shall I fetch it, dear Liza, in what?\n");
    se207_signal(id);

    se207_wait(id);
    printf("There's a hole in the bucket, dear Liza, dear Liza,\n");
    printf("There's a hole in the bucket, dear Liza, a hole. \n");
    se207_signal(id);

  }
}else{
  //P2
  while(1){
    se207_wait(id);
    rsleep();
```

```
    printf("Henry?\n");
    printf("Oh, Henry?\n");
    se207_signal(id);

    se207_wait(id);
    printf("Did you fetch the water?\n");
    se207_signal(id);

    se207_wait(id);
    printf("Well, fix it, dear Henry, dear Henry, dear Henry,\n");
    printf("Well, fix it, dear Henry, dear Henry, fix it.\n");
    se207_signal(id);

    se207_wait(id);
    printf("With a straw, dear Henry, dear Henry, dear Henry,\n");
    printf("With a straw, dear Henry, dear Henry, with a straw.\n");
    se207_signal(id);

    se207_wait(id);
    printf("Cut it, dear Henry, dear Henry, dear Henry,\n");
    printf("Well, cut it, dear Henry, dear Henry, cut it.\n");
    se207_signal(id);

    se207_wait(id);
    printf("With an axe, dear Henry, dear Henry, dear Henry,\n");
    printf("With an axe, dear Henry, dear Henry, an axe.\n");
    se207_signal(id);

    se207_wait(id);
    printf("Sharpen it, dear Henry, dear Henry, dear Henry,\n");
    printf("Well, sharpen it, dear Henry, dear Henry, hone it.\n");
    se207_signal(id);

    se207_wait(id);
    printf("On a stone, dear Henry, dear Henry, dear Henry,\n");
    printf("On a stone, dear Henry, dear Henry, on a stone.\n");
    se207_signal(id);

    se207_wait(id);
    printf("Well, wet it, dear Henry, dear Henry, dear Henry,\n");
    printf("Well, wet it, dear Henry, dear Henry, wet it.\n");
    se207_signal(id);

    se207_wait(id);
    printf("Try water, dear Henry, dear Henry, dear Henry,\n");
    printf("Try water, dear Henry, dear Henry, use water.\n");
    se207_signal(id);
```

```
      se207_wait(id);
      printf("In a bucket, dear Henry, dear Henry, dear Henry,\n");
      printf("In a bucket, dear Henry, dear Henry, in a bucket.\n");
      se207_signal(id);

    }

  }

}
```

Snapshot of the program in action:

```
navickar@hvs-its-lnx01:~/207SE_Sessions/Session15/lab15$ clear
navickar@hvs-its-lnx01:~/207SE_Sessions/Session15/lab15$ ./crit
Semaphore 17960772 initialized with path 'critical_example2.c'.
Sleeping for 5 secs

Sleeping for 5 secs
Henry?
Oh, Henry?
Yes,Liza!
Did you fetch the water?
Don't fetch the water..

There's a hole in the bucket, dear Liza, dear Liza,
There's a hole in the bucket, dear Liza, a hole.
Well, fix it, dear Henry, dear Henry, dear Henry,
Well, fix it, dear Henry, dear Henry, fix it.
With what shall I fix it, dear Liza, dear Liza?
With what shall I fix it, dear Liza, with what?
With a straw, dear Henry, dear Henry, dear Henry,
With a straw, dear Henry, dear Henry, with a straw.
But the straw is too long, dear Liza, dear Liza,
The straw is too long, dear Liza, too long.
Cut it, dear Henry, dear Henry, dear Henry,
Well, cut it, dear Henry, dear Henry, cut it.
With what shall I cut it, dear Liza, dear Liza?
With what shall I cut it, dear Liza, with what?
With an axe, dear Henry, dear Henry, dear Henry,
With an axe, dear Henry, dear Henry, an axe.
The axe is too dull, dear Liza, dear Liza,
The axe is too dull, dear Liza, too dull.
Sharpen it, dear Henry, dear Henry, dear Henry,
Well, sharpen it, dear Henry, dear Henry, hone it.
On what shall I sharpen it, dear Liza, dear Liza?
On what shall I hone it, dear Liza, with what?
On a stone, dear Henry, dear Henry, dear Henry,
On a stone, dear Henry, dear Henry, on a stone.
But the stone is too dry, dear Liza, dear Liza,
The stone is too dry, dear Liza, too dry.
Well, wet it, dear Henry, dear Henry, dear Henry,
Well, wet it, dear Henry, dear Henry, wet it.
With what shall I wet it, dear Liza, dear Liza?
With what shall I wet it, dear Liza, with what?
Try water, dear Henry, dear Henry, dear Henry,
Try water, dear Henry, dear Henry, use water.
In what shall I fetch it, dear Liza, dear Liza?
In what shall I fetch it, dear Liza, in what?
In a bucket, dear Henry, dear Henry, dear Henry,
In a bucket, dear Henry, dear Henry, in a bucket.
There's a hole in the bucket, dear Liza, dear Liza,
There's a hole in the bucket, dear Liza, a hole.
Sleeping for 3 secs
Henry?
Oh, Henry?
Sleeping for 3 secs

Did you fetch the water?
Yes,Liza!
Well, fix it, dear Henry, dear Henry, dear Henry,
Well, fix it, dear Henry, dear Henry, fix it.
```

Advanced Task – Stdout and Stderror:

In order to reduce the file size and processing time, I have limited times of writing to file.

Code:

```c
//critical_example2.c
#include <sys/ipc.h>
#include <sys/sem.h>
#include <stdio.h>
#include <stdlib.h>
```

```c
#include <unistd.h>


#include "se207_sems.h"

int main(int argc, char argv[]){
int id;
//Use our source file as the "key"
  id=se207_semget("critical_example2.c",1);
  int pid=fork();
  FILE *fh = freopen("henry.txt", "w", stdout);
  FILE *fl = fopen("lizy.txt", "w");//open Liza's file

  int i=0, x=0;

  setvbuf(stdout, NULL, _IOLBF, 0);
  if(pid){
    //P1

    while(1&&i<=1){
      se207_wait(id);
      fprintf( fh,"\n");
      se207_signal(id);

      se207_wait(id);
      fprintf( fh,"Yes,Liza!\n");
      se207_signal(id);

      se207_wait(id);
      fprintf( fh,"Don't fetch the water..\n");
      fprintf( fh,"\n");
      fprintf( fh,"There's a hole in the bucket, dear Liza, dear Liza,\n");
      fprintf( fh,"There's a hole in the bucket, dear Liza, a hole.\n");
      se207_signal(id);

      se207_wait(id);
      fprintf( fh,"With what shall I fix it, dear Liza, dear Liza?\n");
      fprintf( fh,"With what shall I fix it, dear Liza, with what?\n");
      se207_signal(id);

      se207_wait(id);
      fprintf( fh,"But the straw is too long, dear Liza, dear Liza,\n");
      fprintf( fh,"The straw is too long, dear Liza, too long.\n");
      se207_signal(id);

      se207_wait(id);
      fprintf( fh,"With what shall I cut it, dear Liza, dear Liza?\n");
      fprintf( fh,"With what shall I cut it, dear Liza, with what?\n");
```

```
      se207_signal(id);

      se207_wait(id);
      fprintf( fh,"The axe is too dull, dear Liza, dear Liza,\n");
      fprintf( fh,"The axe is too dull, dear Liza, too dull.\n");
      se207_signal(id);

      se207_wait(id);
      fprintf( fh,"On what shall I sharpen it, dear Liza, dear Liza?\n");
      fprintf( fh,"On what shall I hone it, dear Liza, with what?\n");
      se207_signal(id);

      se207_wait(id);
      fprintf( fh,"But the stone is too dry, dear Liza, dear Liza,\n");
      fprintf( fh,"The stone is too dry, dear Liza, too dry.\n");
      se207_signal(id);

      se207_wait(id);
      fprintf( fh,"With what shall I wet it, dear Liza, dear Liza?\n");
      fprintf( fh,"With what shall I wet it, dear Liza, with what?\n");
      se207_signal(id);

      se207_wait(id);
      fprintf( fh,"In what shall I fetch it, dear Liza, dear Liza?\n");
      fprintf( fh,"In what shall I fetch it, dear Liza, in what?\n");
      se207_signal(id);

      se207_wait(id);
      fprintf( fh,"There's a hole in the bucket, dear Liza, a hole. \n");
      fprintf( fh,"There's a hole in the bucket, dear Liza, dear Liza,\n");
      se207_signal(id);
      i++;
    }
  }else{
    //P2
    while(1&&x<=1){
      se207_wait(id);
      fprintf(fl,"Henry?\n");
      fprintf(fl,"Oh, Henry?\n");
      se207_signal(id);

      se207_wait(id);
      fprintf(fl,"Did you fetch the water?\n");
      se207_signal(id);

      se207_wait(id);
      fprintf(fl,"Well, fix it, dear Henry, dear Henry, dear Henry,\n");
```

```
    fprintf(fl,"Well, fix it, dear Henry, dear Henry, fix it.\n");
    se207_signal(id);

    se207_wait(id);
    fprintf(fl,"With a straw, dear Henry, dear Henry, dear Henry,\n");
    fprintf(fl,"With a straw, dear Henry, dear Henry, with a straw.\n");
    se207_signal(id);

    se207_wait(id);
    fprintf(fl,"Cut it, dear Henry, dear Henry, dear Henry,\n");
    fprintf(fl,"Well, cut it, dear Henry, dear Henry, cut it.\n");
    se207_signal(id);

    se207_wait(id);
    fprintf(fl,"With an axe, dear Henry, dear Henry, dear Henry,\n");
    fprintf(fl,"With an axe, dear Henry, dear Henry, an axe.\n");
    se207_signal(id);

    se207_wait(id);
    fprintf(fl,"Sharpen it, dear Henry, dear Henry, dear Henry,\n");
    fprintf(fl,"Well, sharpen it, dear Henry, dear Henry, hone it.\n");
    se207_signal(id);

    se207_wait(id);
    fprintf(fl,"On a stone, dear Henry, dear Henry, dear Henry,\n");
    fprintf(fl,"On a stone, dear Henry, dear Henry, on a stone.\n");
    se207_signal(id);

    se207_wait(id);
    fprintf(fl,"Well, wet it, dear Henry, dear Henry, dear Henry,\n");
    fprintf(fl,"Well, wet it, dear Henry, dear Henry, wet it.\n");
    se207_signal(id);

    se207_wait(id);
    fprintf(fl,"Try water, dear Henry, dear Henry, dear Henry,\n");
    fprintf(fl,"Try water, dear Henry, dear Henry, use water.\n");
    se207_signal(id);

    se207_wait(id);
    fprintf(fl,"In a bucket, dear Henry, dear Henry, dear Henry,\n");
    fprintf(fl,"In a bucket, dear Henry, dear Henry, in a bucket.\n");
    se207_signal(id);
    x++;
  }

}
fclose(fl);
```

```
  fclose(fh);
}
```

Henry.txt:

```
navickar@hvs-its-lnx01:~/207SE_Sessions/Session15/lab15$ cat henry.txt

Yes,Liza!
Don't fetch the water..

There's a hole in the bucket, dear Liza, dear Liza,
There's a hole in the bucket, dear Liza, a hole.
With what shall I fix it, dear Liza, dear Liza?
With what shall I fix it, dear Liza, with what?
But the straw is too long, dear Liza, dear Liza,
The straw is too long, dear Liza, too long.
With what shall I cut it, dear Liza, dear Liza?
With what shall I cut it, dear Liza, with what?
The axe is too dull, dear Liza, dear Liza,
The axe is too dull, dear Liza, too dull.
On what shall I sharpen it, dear Liza, dear Liza?
On what shall I hone it, dear Liza, with what?
But the stone is too dry, dear Liza, dear Liza,
The stone is too dry, dear Liza, too dry.
With what shall I wet it, dear Liza, dear Liza?
With what shall I wet it, dear Liza, with what?
In what shall I fetch it, dear Liza, dear Liza?
In what shall I fetch it, dear Liza, in what?
There's a hole in the bucket, dear Liza, a hole.
There's a hole in the bucket, dear Liza, dear Liza,

Yes,Liza!
Don't fetch the water..

There's a hole in the bucket, dear Liza, dear Liza,
There's a hole in the bucket, dear Liza, a hole.
With what shall I fix it, dear Liza, dear Liza?
With what shall I fix it, dear Liza, with what?
But the straw is too long, dear Liza, dear Liza,
The straw is too long, dear Liza, too long.
With what shall I cut it, dear Liza, dear Liza?
With what shall I cut it, dear Liza, with what?
The axe is too dull, dear Liza, dear Liza,
The axe is too dull, dear Liza, too dull.
On what shall I sharpen it, dear Liza, dear Liza?
On what shall I hone it, dear Liza, with what?
But the stone is too dry, dear Liza, dear Liza,
The stone is too dry, dear Liza, too dry.
With what shall I wet it, dear Liza, dear Liza?
With what shall I wet it, dear Liza, with what?
In what shall I fetch it, dear Liza, dear Liza?
In what shall I fetch it, dear Liza, in what?
There's a hole in the bucket, dear Liza, a hole.
There's a hole in the bucket, dear Liza, dear Liza,
navickar@hvs-its-lnx01:~/207SE_Sessions/Session15/lab15$
```

Lizy.txt:

```
navickar@hvs-its-lnx01:~/207SE_Sessions/Session15/lab15$ cat lizy.txt
Henry?
Oh, Henry?
Did you fetch the water?
Well, fix it, dear Henry, dear Henry, dear Henry,
Well, fix it, dear Henry, dear Henry, fix it.
With a straw, dear Henry, dear Henry, dear Henry,
With a straw, dear Henry, dear Henry, with a straw.
Cut it, dear Henry, dear Henry, dear Henry,
Well, cut it, dear Henry, dear Henry, cut it.
With an axe, dear Henry, dear Henry, dear Henry,
With an axe, dear Henry, dear Henry, an axe.
Sharpen it, dear Henry, dear Henry, dear Henry,
Well, sharpen it, dear Henry, dear Henry, hone it.
On a stone, dear Henry, dear Henry, dear Henry,
On a stone, dear Henry, dear Henry, on a stone.
Well, wet it, dear Henry, dear Henry, dear Henry,
Well, wet it, dear Henry, dear Henry, wet it.
Try water, dear Henry, dear Henry, dear Henry,
Try water, dear Henry, dear Henry, use water.
In a bucket, dear Henry, dear Henry, dear Henry,
In a bucket, dear Henry, dear Henry, in a bucket.
Henry?
Oh, Henry?
Did you fetch the water?
Well, fix it, dear Henry, dear Henry, dear Henry,
Well, fix it, dear Henry, dear Henry, fix it.
With a straw, dear Henry, dear Henry, dear Henry,
With a straw, dear Henry, dear Henry, with a straw.
Cut it, dear Henry, dear Henry, dear Henry,
Well, cut it, dear Henry, dear Henry, cut it.
With an axe, dear Henry, dear Henry, dear Henry,
With an axe, dear Henry, dear Henry, an axe.
Sharpen it, dear Henry, dear Henry, dear Henry,
Well, sharpen it, dear Henry, dear Henry, hone it.
On a stone, dear Henry, dear Henry, dear Henry,
On a stone, dear Henry, dear Henry, on a stone.
Well, wet it, dear Henry, dear Henry, dear Henry,
Well, wet it, dear Henry, dear Henry, wet it.
Try water, dear Henry, dear Henry, dear Henry,
Try water, dear Henry, dear Henry, use water.
In a bucket, dear Henry, dear Henry, dear Henry,
In a bucket, dear Henry, dear Henry, in a bucket.
navickar@hvs-its-lnx01:~/207SE_Sessions/Session15/lab15$
```

LAB 16 – Producer and Consumer:

Basic Task:

Description of the producer/consumer problem:

The Producer/Consumer problem occurs when two processes are trying to access the same exact source at the same time. This can only mean one thing – the first to try – gets the data, another one fails. This happens because same resource cannot be accessed twice at the same time.

Advanced Task:

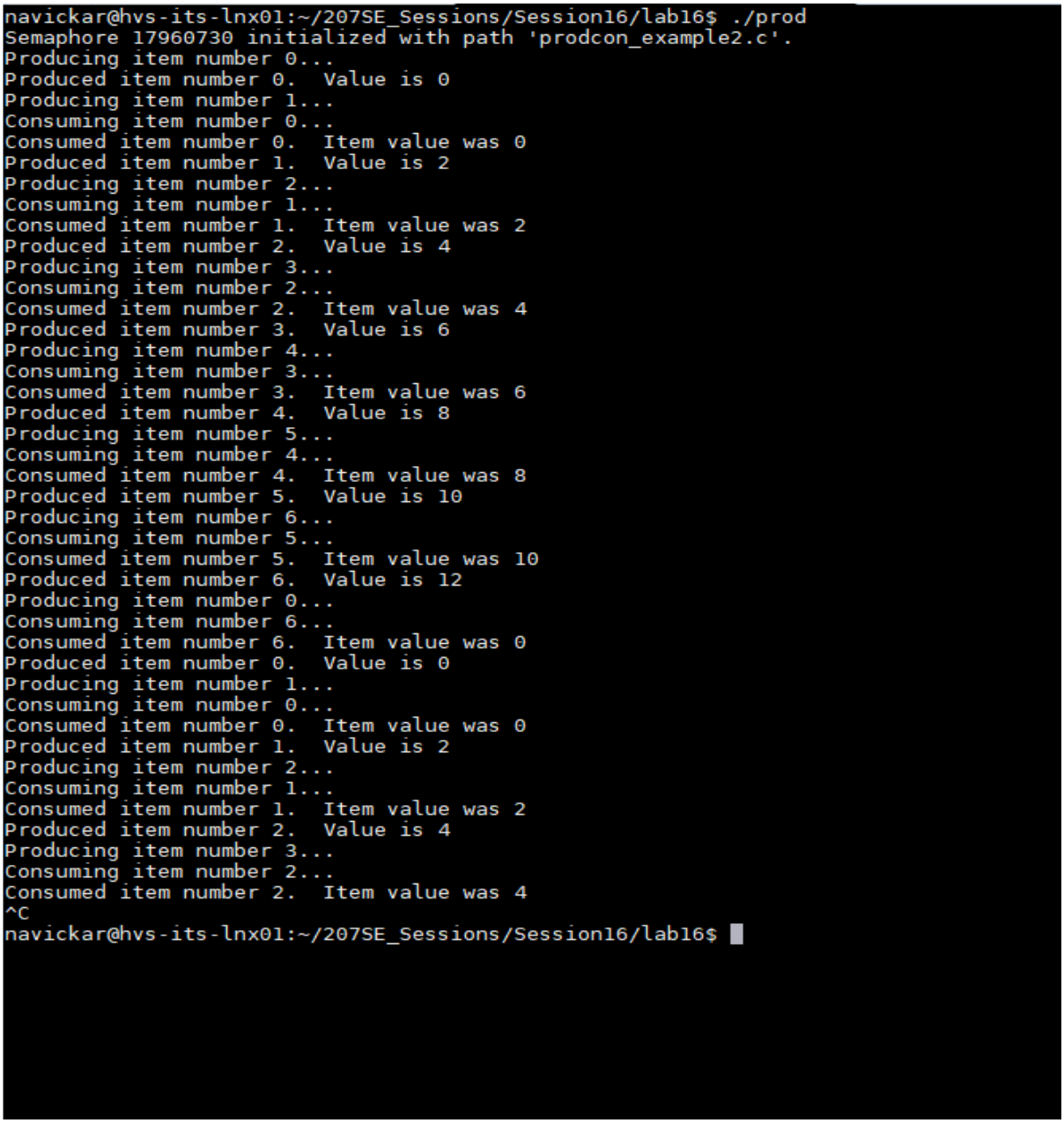
Commented code and evidence of it running:

```
a) #include <sys/ipc.h>
b) #include <sys/sem.h>
c) #include <sys/shm.h>
d) #include <stdio.h>
e) #include <stdlib.h>
f) #include <unistd.h>
g) #include "se207_sems.h"
h) #include <sys/wait.h>
i) /* Remember to try reversing the timings...*/
j)
k) int bufferlength=8; //Limited buffer length
l) //what could we do about this?
m)
n) int main(int argc, char argv[]){
o)
p) pid_t pid;
q)     int status = 0;
r)     int i;
s)   //Create shared memory segment
t)   int shm_id=shmget(ftok("prodcon_example2.c",2),bufferlength,
u)         0666|IPC_CREAT);
v)
w)   //Use our source file as the "key"
x)   int id=se207_semget("prodcon_example2.c",0);
y)
z)   char* data; //For our pointer to shared memory...
aa)
bb)  pid=fork();
cc)  if(pid){
dd)    //P1 - CONSUMER
ee)    shm_id=shmget(ftok("prodcon_example2.c",2),0,006);
ff)
gg)    //Attach the shared buffer
hh)    data = shmat(shm_id, (void *)0, 0);
ii)    int consumed=0;
```

```
jj)     while(consumed<bufferlength){
kk)       if(consumed>=bufferlength-1){//if buffer reached or exceeded the
   length of the buffer, it resets the position so it can continue working
ll)         consumed=0;
mm)       }
nn)       data[bufferlength-1]=consumed;//stores the current location of
   consumer to a last position of buffer
oo)       if (data[bufferlength]-1 != data[bufferlength-2]){//if the
   consumer is not exceeding the limits - proceed as normal.
pp)         se207_wait(id);
qq)         printf("Consuming item number %d...\n",consumed);
rr)         sleep(1);
ss)         char item=data[consumed];
tt)
uu)         printf("Consumed item number %d.  Item value was %d\n",
vv)         consumed,item);
ww)         consumed++;
xx)       }
yy)       else{
zz)         //If the consumer is going to exceed the producer, it will do
   nothing until producer is pushed.
aaa)           }
bbb)         }
ccc)
ddd)         //Detatch
eee)         shmdt(data);
fff)         printf("All done consuming.\n");
ggg)
hhh)         wait(&status); //For child process so that we can
iii)
jjj)         //Delete the shared memory
kkk)         printf("Child ended, removing shm\n");
lll)         shmctl(shm_id, IPC_RMID, NULL);
mmm)       }else{
nnn)         //P2
ooo)         shm_id=shmget(ftok("prodcon_example2.c",2),0,006);
ppp)         //Attach the shared buffer
qqq)         data = shmat(shm_id, (void *)0, 0);
rrr)
sss)         int produced=0;
ttt)         while(produced<bufferlength){
uuu)           if(produced>=bufferlength-1){//If the buffer reached or
   exceeds the limits of the lenght, reset the buffer so it can work
   again.
vvv)             produced=0;
www)           }
xxx)           data[bufferlength-2]=produced;
```

```
yyy)           if (data[bufferlength]-1 != data[bufferlength-2]){//If the
   lenght of the buffer is fine, work as normal.
zzz)             printf("Producing item number %d...\n",produced);
aaaa)            sleep(2);
bbbb)            data[produced]=produced*2; //Simple data, easy to check.
cccc)            printf("Produced item number %d.  Value is %d\n",
dddd)            produced,data[produced]);
eeee)            se207_signal(id);
ffff)            produced++;
gggg)          }
hhhh)          else{
iiii)            //if the producer is exceeding the consumer, it will do
   nothing until consumer is pushed.
jjjj)          }
kkkk)        }
llll)        //Detatch
mmmm)        shmdt(data);
nnnn)        printf("Producer finished.");
oooo)      }
pppp)    }
qqqq)
```

Evidence of program running:



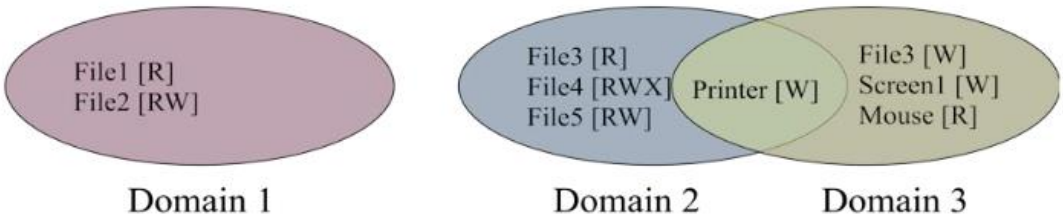
LAB 17 – Client Server:

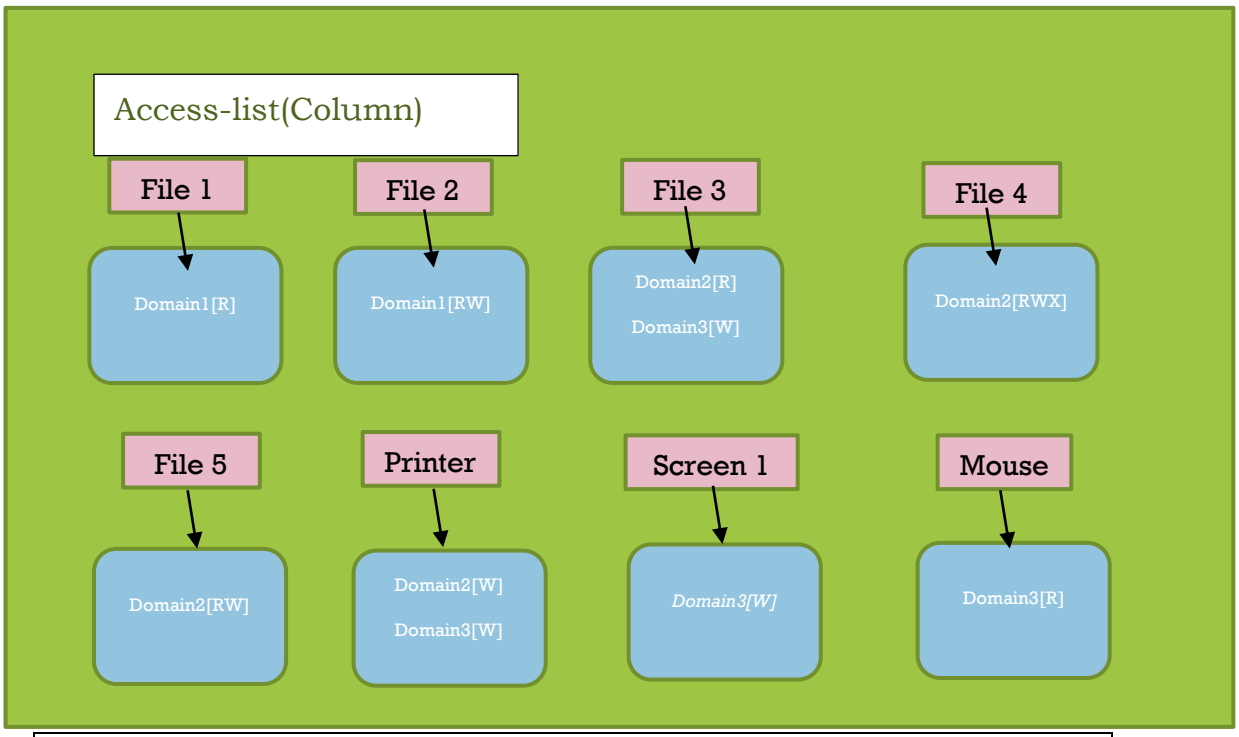
Basic Task:

Advanced Task:

LAB 19 – Operating Systems Security:

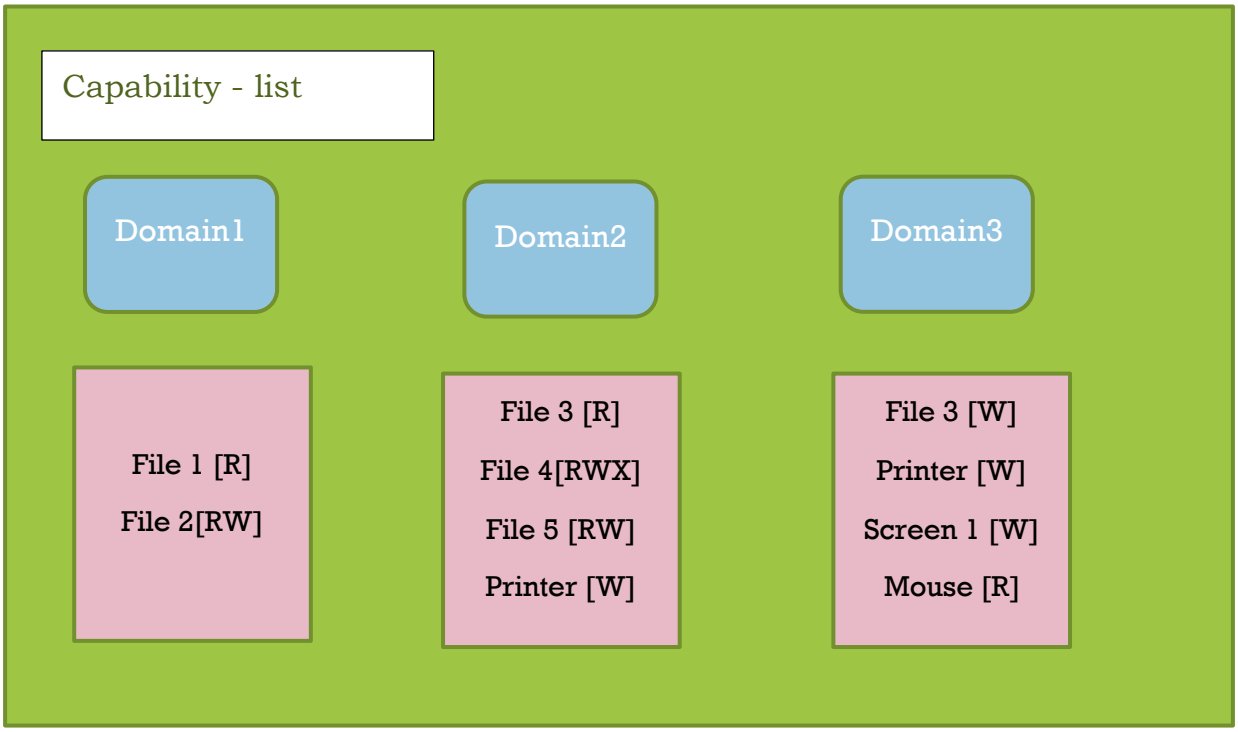
Basic Task:





Protection Domain matrix								
	File1	File2	File3	File4	File5	Printer	Screen1	Mouse
Domain 1	R	RW						
Domain 2			R	RWX	RW	W		
Domain 3			W			W	W	R

Rokas Navickas
207SE – Operating Systems, Security and Networks
Portfolio 2
Student ID: 7868629



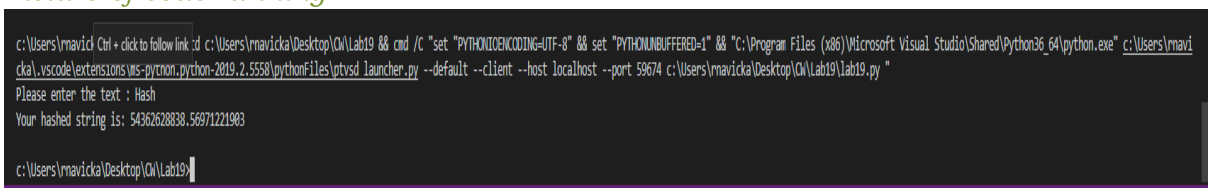
Advanced Task:

Salted Hash:

Explanation of the code:

The reason why I think this hashing code is convenient, is it has double hash function +double salt. It takes the string, converts it to ascii value symbol by symbol , but before that is takes a current timestamp adds as first character, then puts the string characters into a list, after that, takes the list and multiplies first number by five, next by 10, next by 15, etc. Then it takes changed list, adds all the numbers, multiplies by 7 and adds the salt of current date in format of dayYeayMonth(which is quite unusual) and then prints as string.

Picture of code running



Code:

```
import time

#asciival converter, takes string and converts to ascii value
def asciival(str):
```

Rokas Navickas
207SE – Operating Systems, Security and Networks
Portfolio 2
Student ID: 7868629

```
    hashAscii=[]
    hashAscii.append(time.time())#kind of double hashing by adding timestamp
at the beggining
    for s in str:
        x=ord(s) # Ascii value
        hashAscii.append(x) # add to list
    changeVar(hashAscii)

#multiply function, multiple it by a specific number based on its position in
the sequence
def changeVar(hashAscii):
    Counter=5
    integer=0
    while integer<len(hashAscii):
        hashAscii[integer]=hashAscii[integer]*Counter
        Counter+=5
        integer+=1
    salt(hashAscii)

#salt function, sums values, doubles them  and adds salt by the layout of
current time of dayYearMonth
def salt(hashAscii):
    total=sum(hashAscii)
    total=str(total*7)+str(time.strftime("%d%y%m"))
    print("Your hashed string is: " +  total)
    return(total)

#Runs all the hashing process by calling other functions and returning hashed
text
def doHash(string):
    return asciival(string)#Takes the text to ascii format

string=input("Please enter the text : ")
string=list(string)
doHash(string)
```

References

Bridgett, S. (2013, April 15). *Screen, nohup and disown*. Retrieved from Thoughts on a Rainy Day: https://stephenbridgett.wordpress.com/2013/04/15/screen-nohup-and-disown/

Rems, T. (1991). *https://linux.die.net/man/1/watch*. Retrieved from watch(1) - Linux man page: https://linux.die.net/man/1/watch