SYSTEM PROGRAMMING

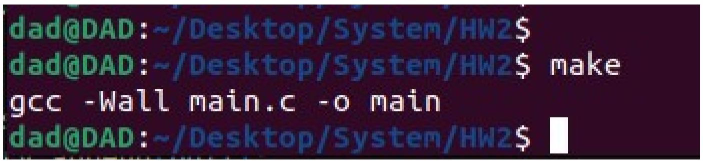
HOMEWORK 2

Derviş Ali Duman
1801042626

SOLUTION APPROACHES

I started to take commands in while loop.
Parsed them so I can see the which command should applied.
Looked for :q command so I can terminate the program.
I tokenized the commands so each command will run in different child process.
Each tokenized command will execute in execvp and they will written on pipe.
When other childs tries to take those inputs pipe is now ready to read.
When command executed program gets timestamp and executes the create log file.
Each child pid written in logfile so you can understand they worked in different child.
For signal handling I used sigaction.
When terminator signal comes then all child process will terminate.

COMPILATION

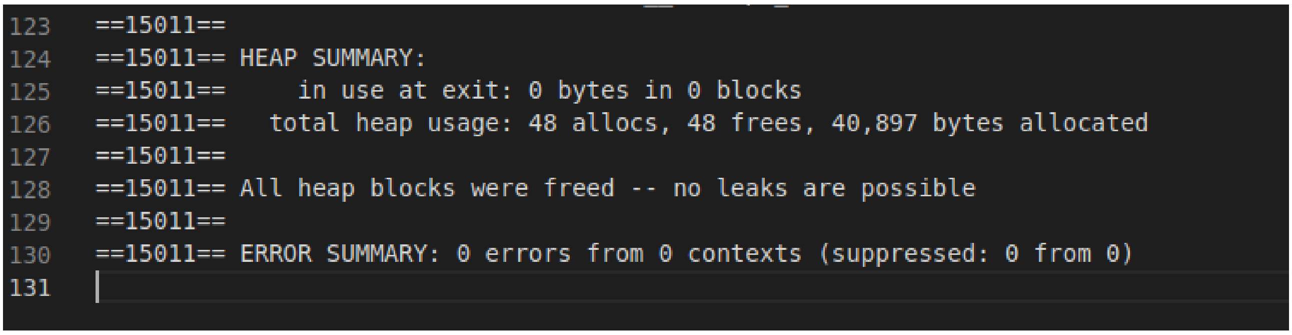


```
dad@DAD:~/Desktop/System/HW2$
dad@DAD:~/Desktop/System/HW2$ make
gcc -Wall main.c -o main
dad@DAD:~/Desktop/System/HW2$ 
```

EXECUTING

./main

VALGRIND

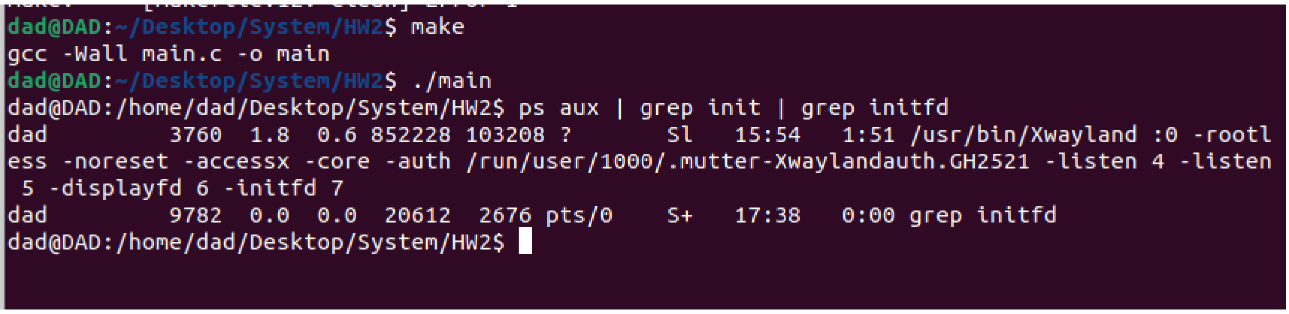


```
==15011==
==15011== HEAP SUMMARY:
==15011==     in use at exit: 0 bytes in 0 blocks
==15011==   total heap usage: 48 allocs, 48 frees, 40,897 bytes allocated
==15011==
==15011== All heap blocks were freed -- no leaks are possible
==15011==
==15011== ERROR SUMMARY: 0 errors from 0 contexts (suppressed: 0 from 0)
```

OUTPUTS

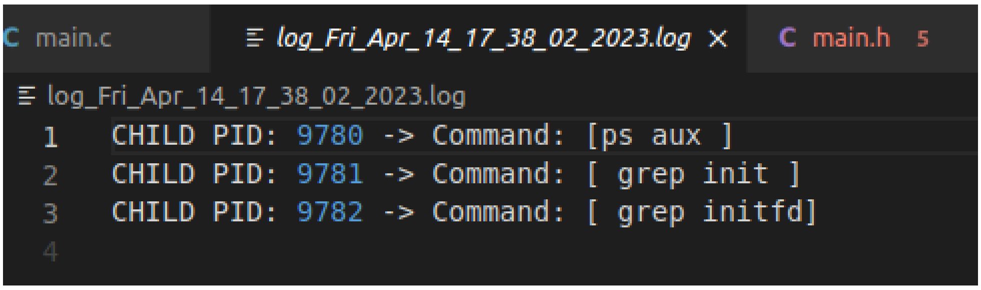
PART 1

IN PDF: Each shell command should be executed via a newly created child process, meaning that multiple commands will result in multiple child processes.



```
dad@DAD:~/Desktop/System/HW2$ make
gcc -Wall main.c -o main
dad@DAD:~/Desktop/System/HW2$ ./main
dad@DAD:/home/dad/Desktop/System/HW2$ ps aux | grep init | grep initfd
dad         3760  1.8  0.6 852228 103208 ?       Sl   15:54   1:51 /usr/bin/Xwayland :0 -rootl
ess -noreset -accessx -core -auth /run/user/1000/.mutter-Xwaylandauth.GH2521 -listen 4 -listen
 5 -displayfd 6 -initfd 7
dad         9782  0.0  0.0  20612  2676 pts/0    S+   17:38   0:00 grep initfd
dad@DAD:/home/dad/Desktop/System/HW2$
```

Log:

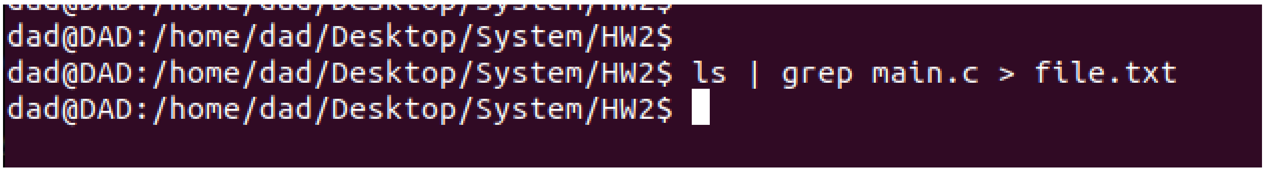

C main.c
log_Fri_Apr_14_17_38_02_2023.log ×
C main.h 5
log_Fri_Apr_14_17_38_02_2023.log

```
CHILD PID: 9780 -> Command: [ps aux ]
CHILD PID: 9781 -> Command: [ grep init ]
CHILD PID: 9782 -> Command: [ grep initfd]
```

As you can see there are 3 process created.

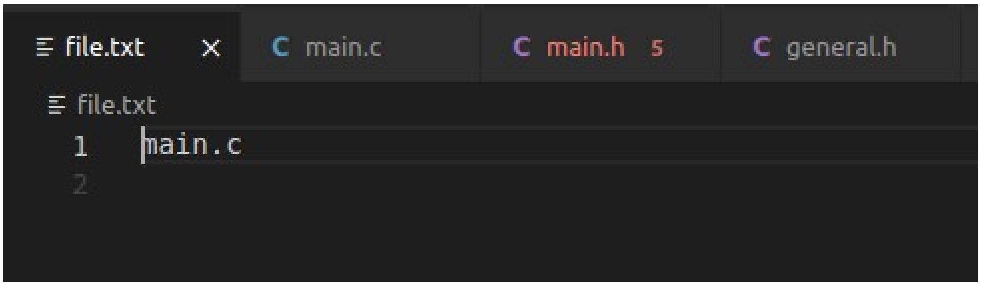
PART 2

IN PDF: Proper handling of pipes ("|") and redirections ("<", ">") by redirecting the appropriate file descriptors.



```
dad@DAD:/home/dad/Desktop/System/HW2$
dad@DAD:/home/dad/Desktop/System/HW2$ ls | grep main.c > file.txt
dad@DAD:/home/dad/Desktop/System/HW2$
```

Output file:



```
main.c
```

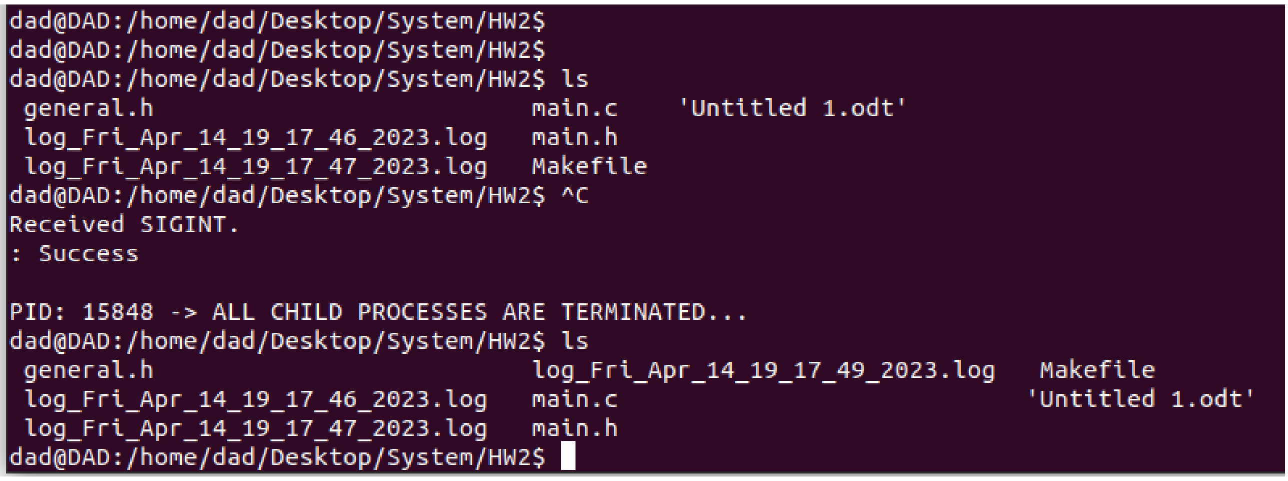
PART 3

IN PDF: Usage information should be printed if the program is not called properly.

You will just execute the code with ./main so you don’t need it.

PART 4

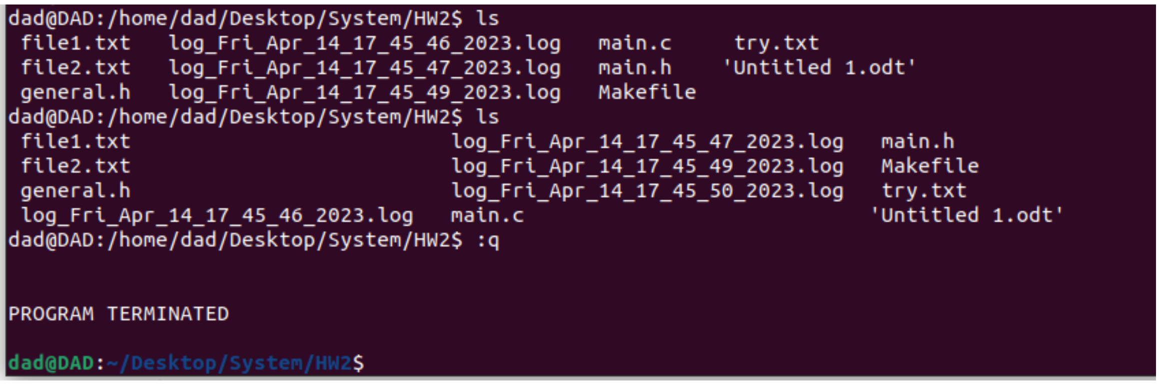
IN PDF: Error messages and signals that occur during execution should be printed, and the program should return to the prompt to receive new commands.



```
dad@DAD:/home/dad/Desktop/System/HW2$
dad@DAD:/home/dad/Desktop/System/HW2$
dad@DAD:/home/dad/Desktop/System/HW2$ ls
 general.h                           main.c     'Untitled 1.odt'
 log_Fri_Apr_14_19_17_46_2023.log    main.h
 log_Fri_Apr_14_19_17_47_2023.log    Makefile
dad@DAD:/home/dad/Desktop/System/HW2$ ^C
Received SIGINT.
: Success

PID: 15848 -> ALL CHILD PROCESSES ARE TERMINATED...
dad@DAD:/home/dad/Desktop/System/HW2$ ls
 general.h                           log_Fri_Apr_14_19_17_49_2023.log   Makefile
 log_Fri_Apr_14_19_17_46_2023.log    main.c                            'Untitled 1.odt'
 log_Fri_Apr_14_19_17_47_2023.log    main.h
dad@DAD:/home/dad/Desktop/System/HW2$ 
```

PART 5

IN PDF: Aside from a SIGKILL (which also should be handled properly) the program must wait for “:q” to finalize its execution.

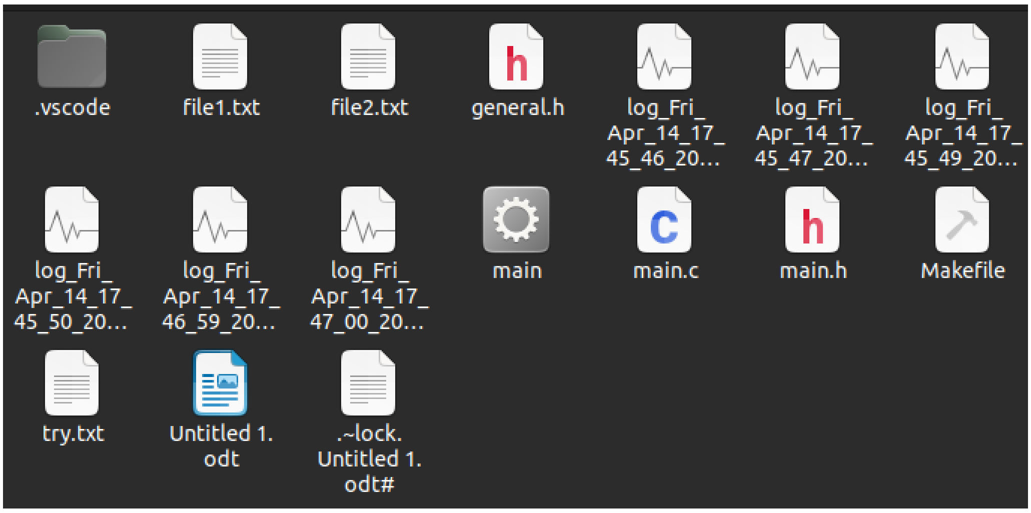


```
dad@DAD:/home/dad/Desktop/System/HW2$ ls
 file1.txt   log_Fri_Apr_14_17_45_46_2023.log   main.c     try.txt
 file2.txt   log_Fri_Apr_14_17_45_47_2023.log   main.h    'Untitled 1.odt'
 general.h   log_Fri_Apr_14_17_45_49_2023.log   Makefile
dad@DAD:/home/dad/Desktop/System/HW2$ ls
 file1.txt                          log_Fri_Apr_14_17_45_47_2023.log   main.h
 file2.txt                          log_Fri_Apr_14_17_45_49_2023.log   Makefile
 general.h                          log_Fri_Apr_14_17_45_50_2023.log   try.txt
 log_Fri_Apr_14_17_45_46_2023.log   main.c                            'Untitled 1.odt'
dad@DAD:/home/dad/Desktop/System/HW2$ :q


PROGRAM TERMINATED

dad@DAD:~/Desktop/System/HW2$
```

PART 6

IN PDF: Upon completion, all pids of child processes with their corresponding commands should be logged in a separate file. Each execution should create a new log file with a name corresponding to the current timestamp

log files:


.vscode
file1.txt
file2.txt
general.h
log_Fri_ Apr_14_17_ 45_46_20…
log_Fri_ Apr_14_17_ 45_47_20…
log_Fri_ Apr_14_17_ 45_49_20…
log_Fri_ Apr_14_17_ 45_50_20…
log_Fri_ Apr_14_17_ 46_59_20…
log_Fri_ Apr_14_17_ 47_00_20…
main
main.c
main.h
Makefile
try.txt
Untitled 1. odt
.~lock. Untitled 1. odt#