

# CS314: Operating Systems Laboratory

## Laboratory – 2

Hemanth Surya S (180010018)

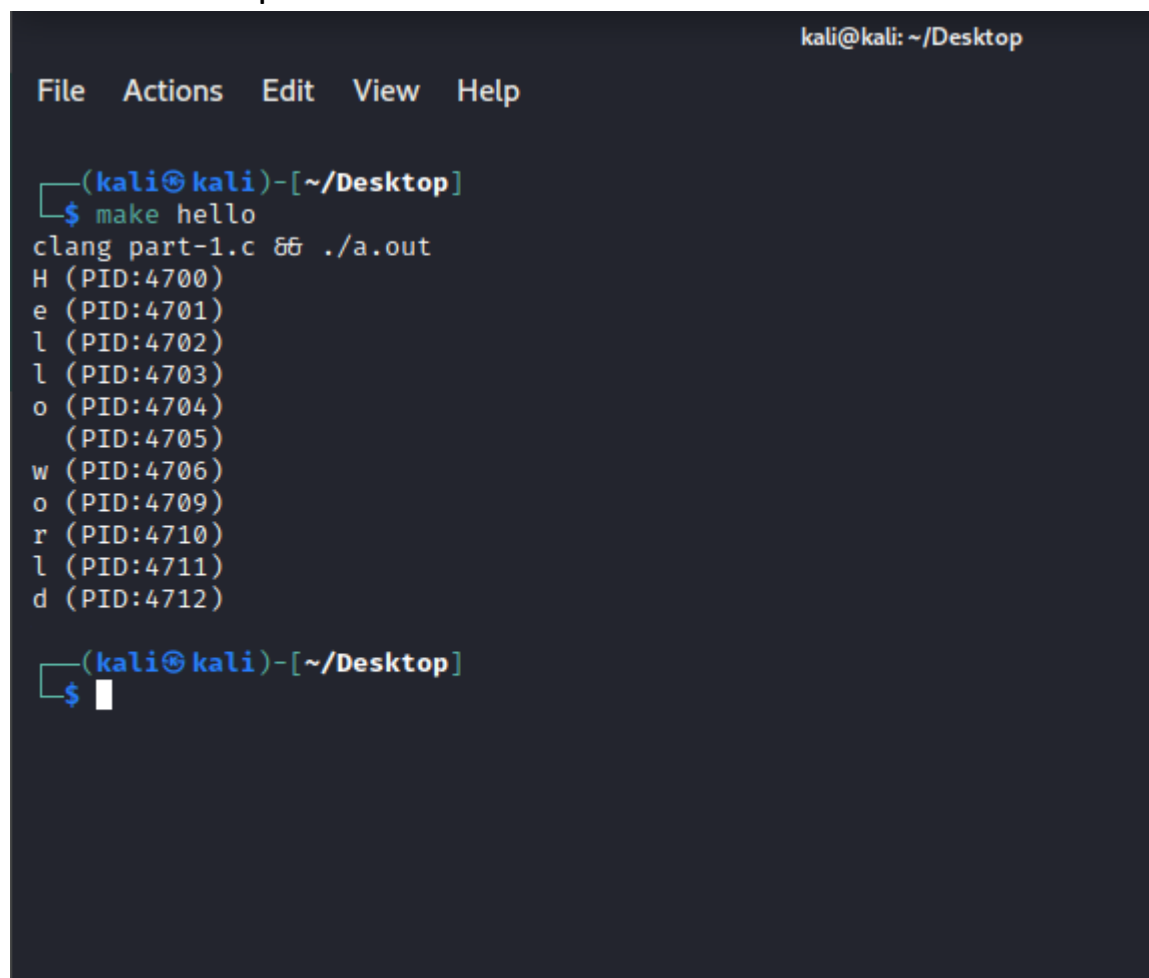
### Part-1:

#### Implementation:

1. Hello world is stored as a string.
2. The parent process is made to wait by letting child process to execute ;print each character of string, sleep for some random interval between 1-4 secs and fork() iteratively.

The task can be achieved with as low as 12 lines of C code.

Here's the output.



```
kali@kali: ~/Desktop
File Actions Edit View Help
(kali@kali)-[~/Desktop]
$ make hello
clang part-1.c -o ./a.out
H (PID:4700)
e (PID:4701)
l (PID:4702)
l (PID:4703)
o (PID:4704)
(PID:4705)
w (PID:4706)
o (PID:4709)
r (PID:4710)
l (PID:4711)
d (PID:4712)
(kali@kali)-[~/Desktop]
$
```

## Part-2:

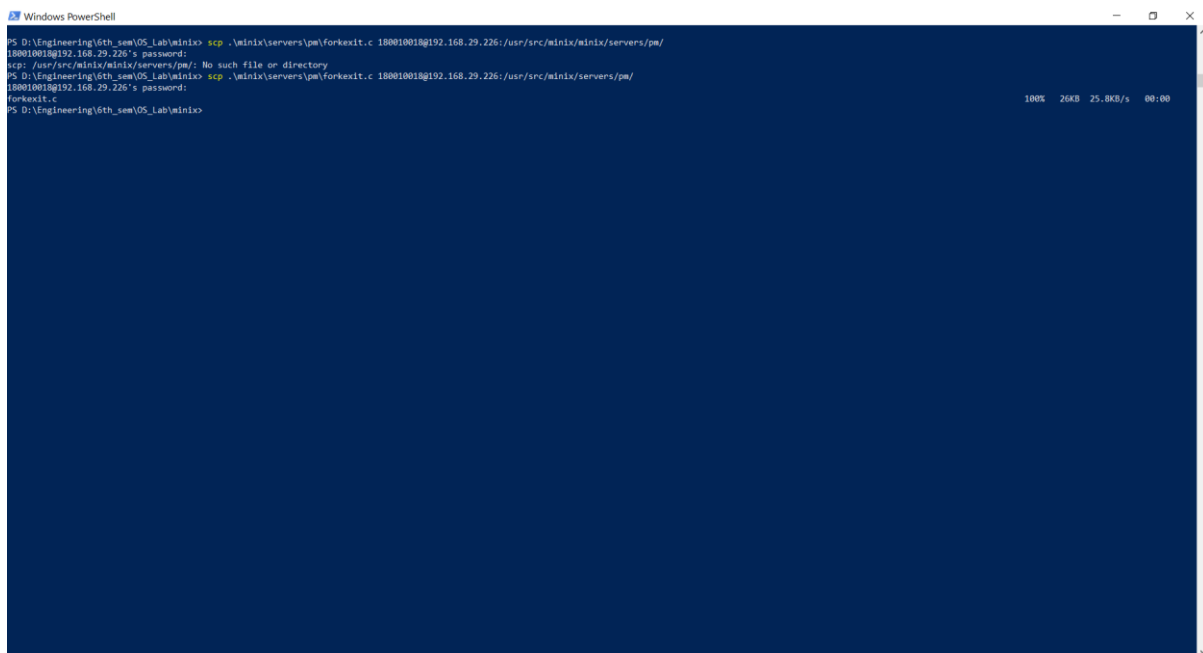
When referred to forkexit.c file, in do\_fork when assigning pid to child, we print the pid as directed i.e.,

“Minix: PID <pid> created”

In do\_exit, after the process completion, we print the below statement as we dump the core.

"Minix: PID <pid> exited"

The above changes made is transferred to minix VM using scp command.



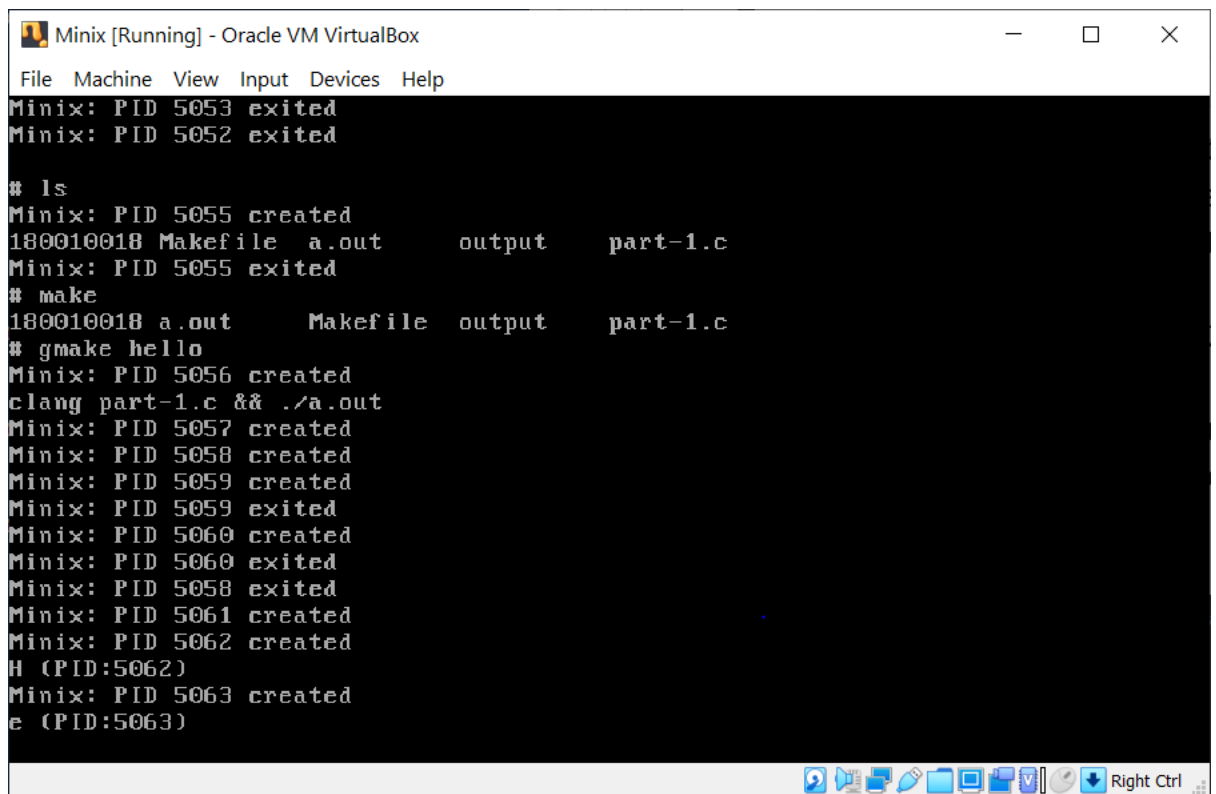
```
Windows PowerShell
PS D:\Engineering\6th_sem\OS_Lab\minix> scp .\minix\servers\pm\forkexit.c 10001001@192.168.29.226:/usr/src/minix/minix/servers/pm/
10001001@192.168.29.226's password:
scp: /usr/src/minix/minix/servers/pm/: No such file or directory
PS D:\Engineering\6th_sem\OS_Lab\minix> scp .\minix\servers\pm\forkexit.c 10001001@192.168.29.226:/usr/src/minix/servers/pm/
10001001@192.168.29.226's password:
forkexit.c
PS D:\Engineering\6th_sem\OS_Lab\minix>
```

Fig-3

Run run.sh to build:

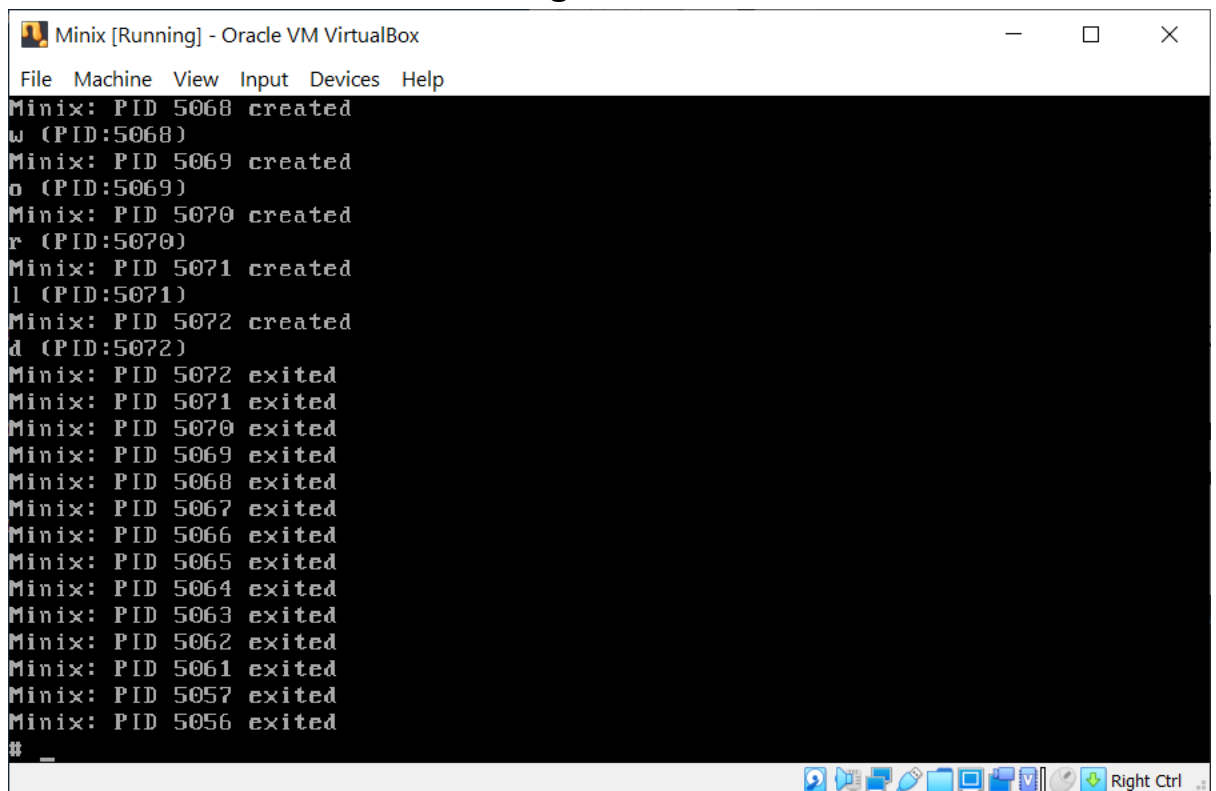
- make build MKUPDATE=yes
- reboot

Changes made to minix VM can now be used to justify part-1 correctness. PIDs are printed beside each character.



```
Minix [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
Minix: PID 5053 exited
Minix: PID 5052 exited
# ls
Minix: PID 5055 created
180010018 Makefile a.out      output      part-1.c
Minix: PID 5055 exited
# make
180010018 a.out      Makefile output      part-1.c
# gmake hello
Minix: PID 5056 created
clang part-1.c && ./a.out
Minix: PID 5057 created
Minix: PID 5058 created
Minix: PID 5059 created
Minix: PID 5059 exited
Minix: PID 5060 created
Minix: PID 5060 exited
Minix: PID 5058 exited
Minix: PID 5061 created
Minix: PID 5062 created
H (PID:5062)
Minix: PID 5063 created
e (PID:5063)
```

Fig-1



```
Minix [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
Minix: PID 5068 created
w (PID:5068)
Minix: PID 5069 created
o (PID:5069)
Minix: PID 5070 created
r (PID:5070)
Minix: PID 5071 created
l (PID:5071)
Minix: PID 5072 created
d (PID:5072)
Minix: PID 5072 exited
Minix: PID 5071 exited
Minix: PID 5070 exited
Minix: PID 5069 exited
Minix: PID 5068 exited
Minix: PID 5067 exited
Minix: PID 5066 exited
Minix: PID 5065 exited
Minix: PID 5064 exited
Minix: PID 5063 exited
Minix: PID 5062 exited
Minix: PID 5061 exited
Minix: PID 5057 exited
Minix: PID 5056 exited
#
```

Fig-2

The Process which is printing 'H' has PID: 5062. The process that prints the next character 'e' has PID:5063 and goes on without terminating their parent process. That indicates that each process is child of the previous process. It all gets terminated once in the end.(As seen in the end of second figure)

Fig-4