

Shahil Patel-200010039
Assignment-2
CS-314:Operating Systems Laboratory

Part-1

Lines of the code: 22

```
You, 6 days ago | 1 author (You)
1  #include <stdio.h>
2  #include <stdlib.h>
3  #include <unistd.h>
4  #include <sys/wait.h>
5
6  int main(){
7      char str[] = "Hello World";
8      int pid = 0;
9      for(int i=0; i<11; i++){
10         if(pid == 0){
11             printf("%c (pid = %d)\n", str[i], (int) getpid());
12             // Select an random integer between 1 to 4
13             int n = (rand()%4) + 1;
14             sleep(n);
15
16             if(i != 10){
17                 pid = fork();
18             }
19         }
20     }
21     wait(NULL);
22     return 0;
23 }
24 // Minimum lines of C code: 20
```

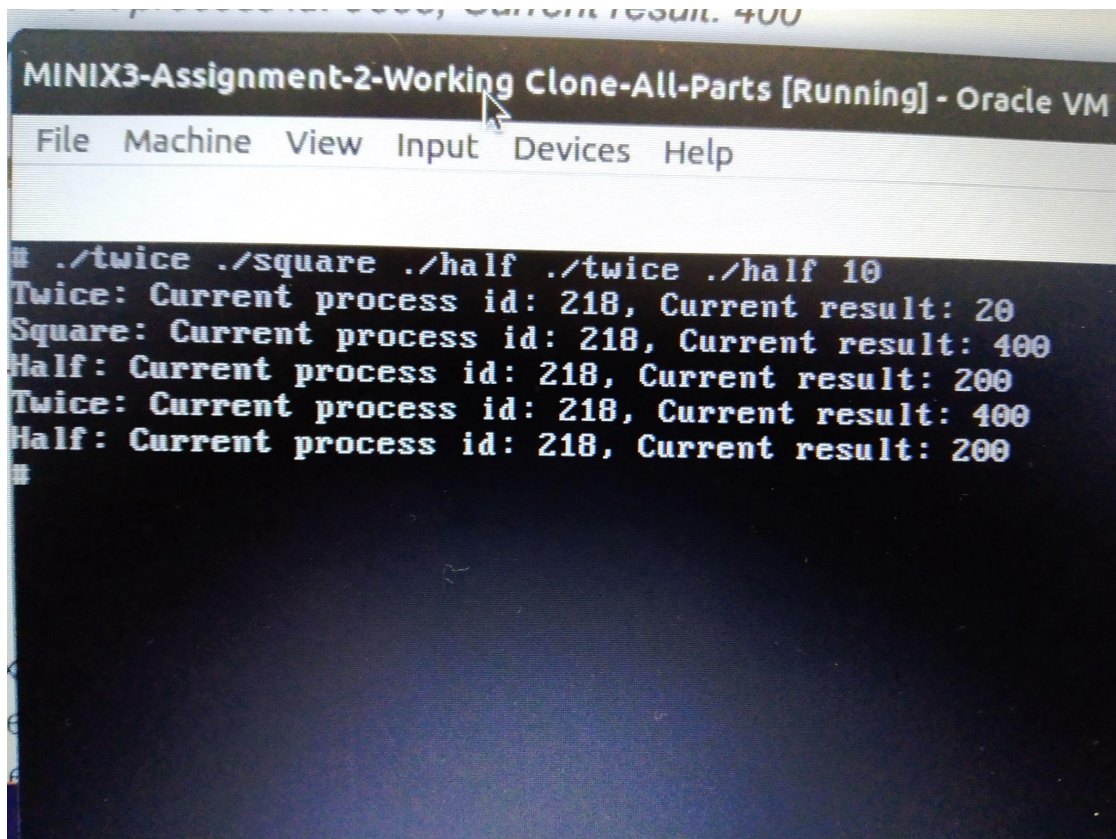
Command to run the file in MINIX3: ***gmake hello.***

The output result is:

```
H Current Process ID (PID) = 222
e Current Process ID (PID) = 223
l Current Process ID (PID) = 224
l Current Process ID (PID) = 225
o Current Process ID (PID) = 226
  Current Process ID (PID) = 227
W Current Process ID (PID) = 228
o Current Process ID (PID) = 229
r Current Process ID (PID) = 230
l Current Process ID (PID) = 231
d Current Process ID (PID) = 232
# _
```

Part-2

The output for the command: `./twice ./square ./half ./twice ./half 10` is:



```
MINIX3-Assignment-2-Working Clone-All-Parts [Running] - Oracle VM
File Machine View Input Devices Help

# ./twice ./square ./half ./twice ./half 10
Twice: Current process id: 218, Current result: 20
Square: Current process id: 218, Current result: 400
Half: Current process id: 218, Current result: 200
Twice: Current process id: 218, Current result: 400
Half: Current process id: 218, Current result: 200
#
```

Part-3

The processes are created in a tree-like fashion where the parent process is created first, which creates a child process, and processes exit in reverse order, the child process exits first, and then the parent process exits.

The file **forkexit.c** was changed at *minix/servers/pm* The following statement was added in forkexit.c file when a process is created:

```
printf("Minix: PID %d created\n", rmc->mp_pid);
```

Similarly, the following statement was added in forkexit.c file when a process exited:

```
printf("Minix: PID %d exited\n", mp->mp_pid);
```

Sample Output:

```
Minix: PID 361 exited
# ls
Minix: PID 362 created
.exrc .profile .ssh
Minix: PID 362 exited
# ls
Minix: PID 363 created
.exrc .profile .ssh
Minix: PID 363 exited
# -
```