Shriram Ghadge
Roll No. 180010015

CS314 - Lab 2

Part I :

`fork()` is used to create a child process. The return value from this, is used to decide whether the current process is a parent or child.

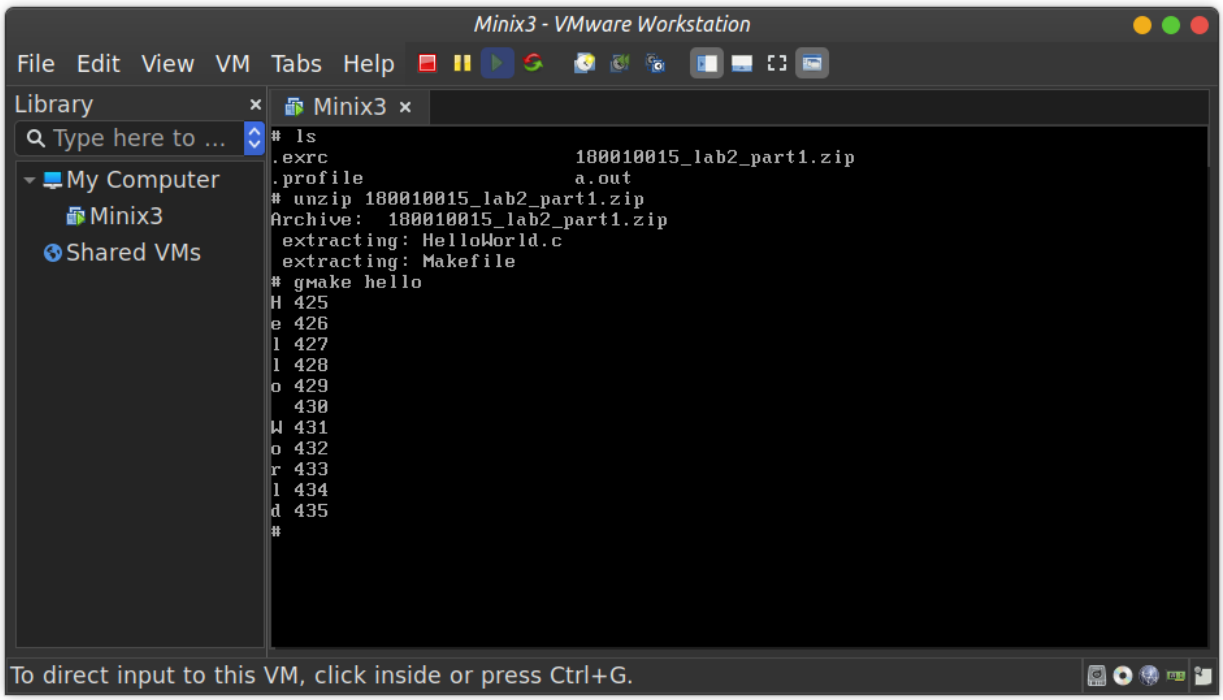
`sleep()` is another useful function from `unistd.h`, stops program execution for particular time limit

`rand()` selects random number.

A function to print ‘hello world’ is called with the string to be printed as argument, its length and index of char to print. Function verifies if the index is less than the string's length then creates a child process. Using id from fork, it checks if it is the parent process.

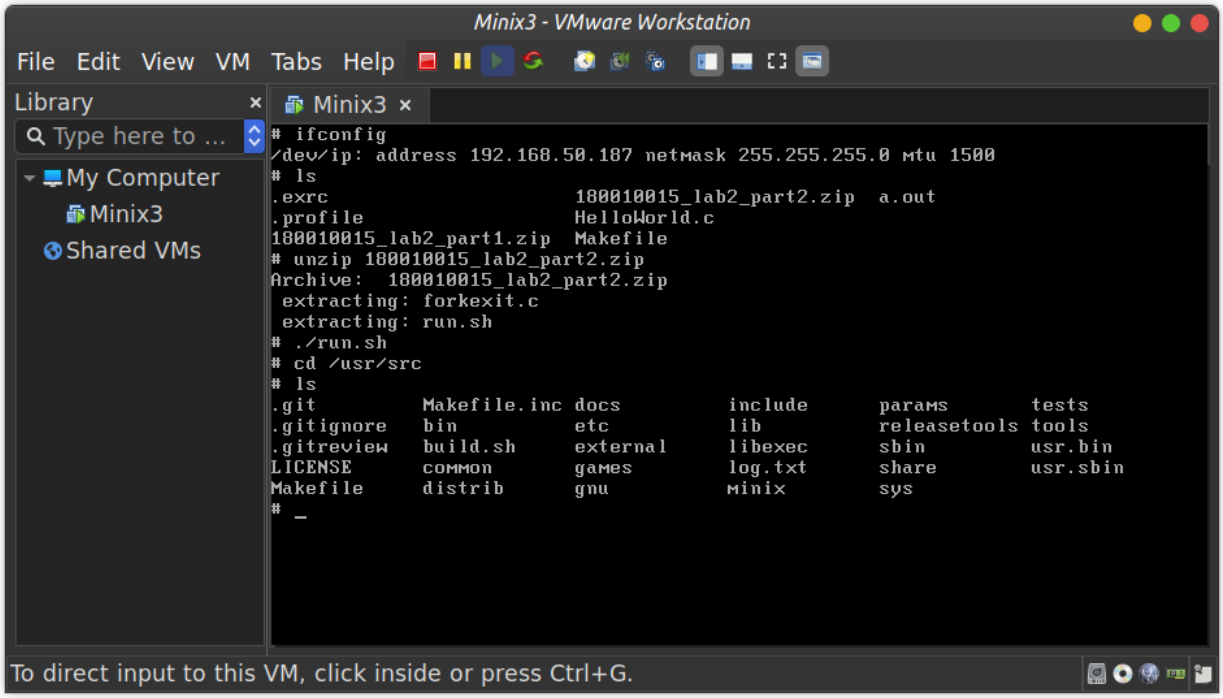
What is the minimum lines of C code with which you can achieve the above?

- For me C code was of around 27 lines. (it can be optimised)

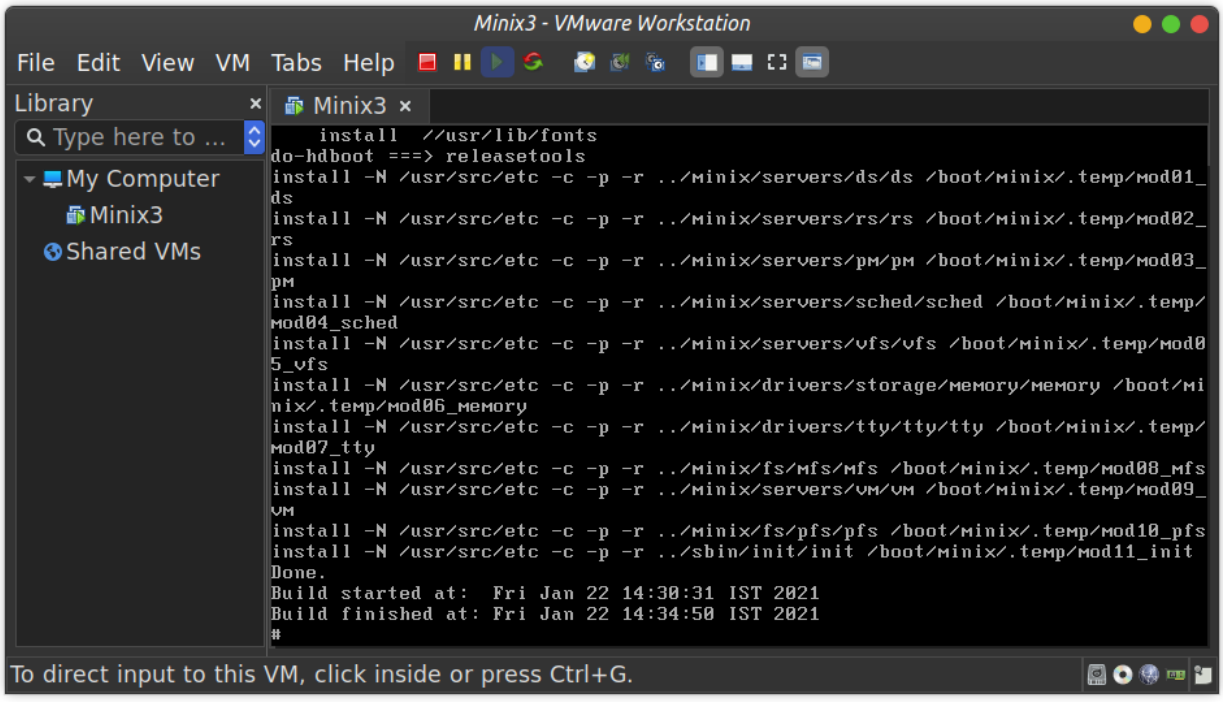


Part II :

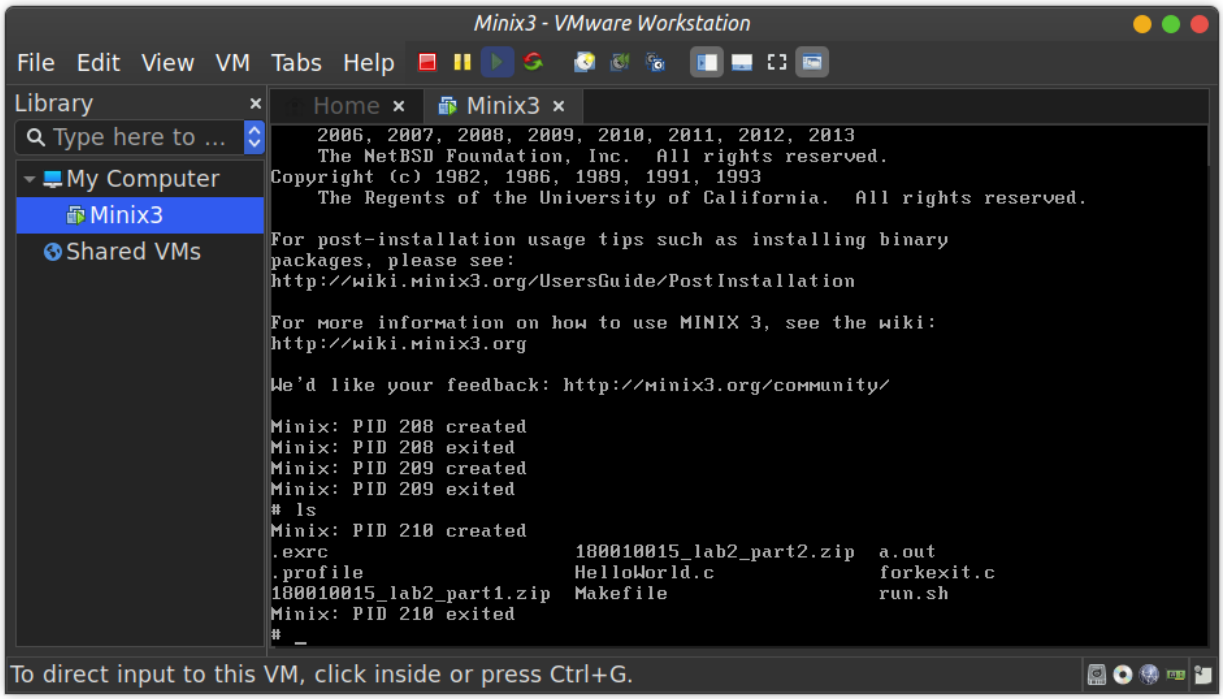
cp forkexit.c /usr/src/minix/servers/pm/forkexit.c



cd /usr/src && make build MKUPDATE=yes >log.txt 2>log.txt
cat log.txt



reboot



Processes have 5 stages :

Start
Ready
Running
Waiting
Terminated

Process creation is sequential, if PID - i was last created process, next will be with PID - i+1. First will be ‘init’ process & subsequent processes will be child processes. Exits of processes need not to be sequential.