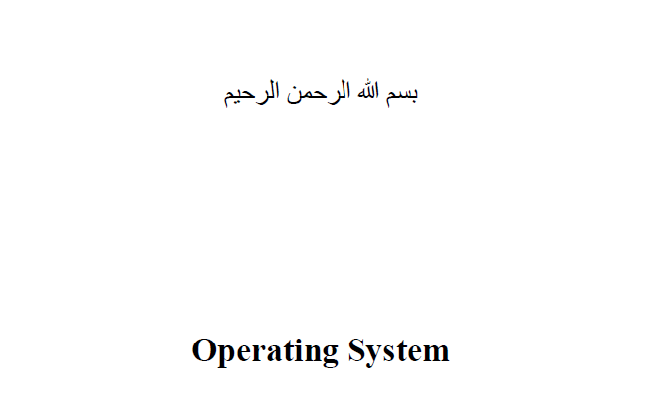
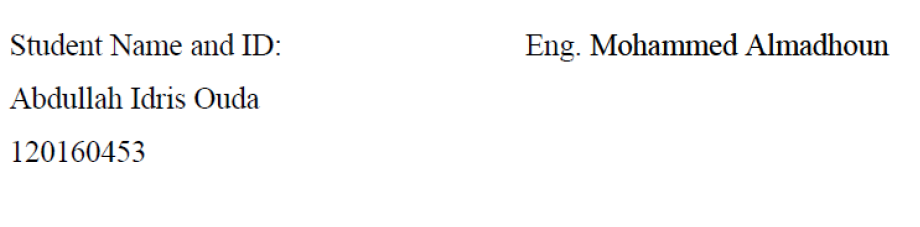
****

**Command Line interpreter Project**

****

**1.** The shell must support the following internal commands:

**i.** cd <directory>—Change the current default directory to <directory>.

If the <directory> argument is not present, report the current

directory. If the directory does not exist, an appropriate error should be

reported. This command should also change the PWD environment

variable. ✅

**ii.** clr—Clear the screen. ✅

**iii.** dir <directory>—List the contents of directory <directory>. ✅

**iv.** environ—List all the environment strings. ✅

**v.** echo <comment>—Display <comment> on the display followed by a new line (multiple spaces/tabs may be reduced to a single space). ✅

**vi.** help—Display the user manual using the more filter. ✅

**vii.** pause—Pause operation of the shell until “Enter” is pressed. ✅

**viii.** quit—Quit the shell. ✅

**ix.** The shell environment should contain shell=<pathname>/myshell

where <pathname>/myshell is the full path for the shell executable (not

a hardwired path back to your directory, but the one from which it was

executed). ✅

**2.** All other command line input is interpreted as program invocation, which

should be done by the shell forking and execing the programs as its own child

processes. The programs should be executed with an environment that contains

the entry: parent=<pathname>/myshell where <pathname>/myshell

is as described in 1.ix above. ✅

**3.** The shell must be able to take its command line input from a file. That is, if the

shell is invoked with a command line argument:

myshell batchfile

then batchfile is assumed to contain a set of command lines for the shell to

process. When the end-of-file is reached, the shell should exit. Obviously, if the

shell is invoked without a command line argument, it solicits input from the user

via a prompt on the display. ❎

**4.** The shell must support I/O redirection on either or both *stdin* and/or *stdout*.

That is, the command line

programname arg1 arg2 < inputfile > outputfile

will execute the program programname with arguments arg1 and arg2, the

*stdin FILE stream* replaced by inputfile and the *stdout FILE stream*

replaced by outputfile.

stdout redirection should also be possible for the internal commands dir,

environ, echo, and help.

With output redirection, if the redirection character is > then the

outputfile

is created if it does not exist, and truncated if it does. If the redirection

token is >> then outputfile is created if it does not exist, and

appended to if it does. ✅

**5.** The shell must support background execution of programs. An ampersand (&) at the end of the command line indicates that the shell should return to the command line prompt immediately after launching that program. ❎

**6.** The command line prompt must contain the pathname of the current

directory. ✅