PA 1 – REPORT

What is the command ping and why did I choose it:

Ping command's man page explanation is this: send ICMP_ECHO request to network hosts. What ping does is basically it tries to reach to the definition that you specify (for example Google.com) from your computer. If it is successful (there is a connection between your computer and the specified address) you will most likely see a response from that address. If it is not successfull you won't get a response.

This summer, my internship made use of Network Address Translation. During the internship I had to use to ping command to test some of my work. Since I used the command frequently I wanted to use it again for this assignment.

What does the option -v do and why did I choose it:

-v command gives verbose output to the user.

The reason for me choosing this option is because I used to use this option almost always with the ping option because it gives more information about what the ping command is doing while trying to reach the destination that you specify.

Process Hierarchy:

First thing the program does is to print out the "I'm SHELL …" line since we are on the main process (parent) so that this will be printed out only once and it will be the first line to be printed out every time we run the code.

Second thing the program does is that it declares a file descriptor in order to do the pipe functionality. Right after declaring it the program checks if there is an error with it or not.

After these operations program makes its first fork call leading up to the creation of the first child (child1) process. After checking if the child1 process was successfully created, child1 process then prints out the "I'm MAN process…" line. Later child1 process uses the dup2(fd[1],STD_OUTFILENO) command. fd[1] is now used for writing the output of this process meaning that whatever comes out of this process as an output it won't be printed out to the terminal but it would be sent straight to the pipe. Therefore here we are making sure that the output of the "man ping" command is going to the pipe rather than our screen (terminal screen). After this operation the program creates a char array containing the commands that we want it to execute which are "man" and "ping". With the use of execvp we execute this command. The child1's work ends here.

Now the program goes back to the parent. First thing the parent does is to wait (waitpid) child1 since child1's operations have to end before creating child2. The parent then creates another child with the second fork call (child2). If child2 is created successfully, it prints the line "I'm GREP process…". After that we do another dup2 call (dup2(fd[0],STD_INFILENO)). In this call we connect the pipe's other end. This call makes this process listen and take input from what comes out of the pipe which is the child1's output. In our case this call gets the output of “man ping” command as input. We then create the output.txt file and give standard output’s output, which is the output that would come out after the child2 process is done, into the file that we’ve created. After that we execute our command “grep -A 1 -e -v” with the use of execvp again. Since this command has input of the pipe it sees ping command’s manual page and executes the command “grep -A 1 -e -v” in that page. However the output isn’t printed onto the terminal. This is because of the operation that we’ve done on child2 which was to direct the output of the child2 process into the file that we specified which is output.txt.

Lastly the program returns to the main process again (parent) and the main process closes both ends of the pipe and waits for the child processes to end in order to print the last statement which is “I’m SHELL … - execution is complete …”.

Ural Sarp Sipahi 28093