

OS Assignment: Building your own shell

3B: add support to execute commands

Assume that the full path to the executable is provided as an input. For example: suppose a C program is compiled into an executable “prog” in directory /home/user, then we want to run that program in the shell as follows:

```
$/home/user/src/prog
```

For this assignment, you need to use the system calls, fork and exec (execv). The shell will follow the pattern shown below:

```
pid = fork();
if (pid == 0) {
    // child process
    exec(cmd, args);
} else {
    // the parent waits for the child
    wait(pid);
}
```

3C: add support for output redirection

```
$/home/sriramk/src/prog > out.txt
```

The syntax for output redirection is [process] > [file]. When you run a program, the output is usually printed on the console (STDOUT stream). But, if you use the redirection operator, the output is redirected to out.txt instead of the console.

The shell can achieve this using the dup2 system call. The STDOUT stream of the child needs to be replaced with the file descriptor corresponding to out.txt.

```
dup2 (fdout, STDOUT_FILENO);
```

Acknowledgments:

Code snippets from UC Berkeley's offering have been used for this assignment.