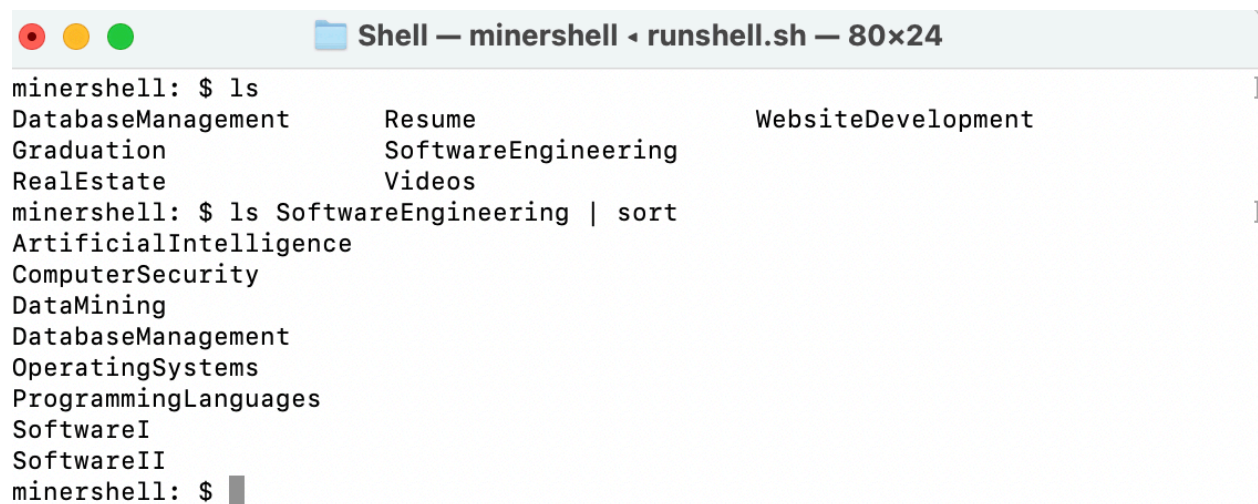


Assignment #2C
Jesus Gutierrez

Augmenting your “Minershell” with pipelining functionality:

In the below example, “sort” command takes the output of “ls <mydirectory>” as its input.



```
minershell: $ ls
DatabaseManagement      Resume                  WebsiteDevelopment
Graduation              SoftwareEngineering
RealEstate               Videos
minershell: $ ls SoftwareEngineering | sort
ArtificialIntelligence
ComputerSecurity
DataMining
DatabaseManagement
OperatingSystems
ProgrammingLanguages
SoftwareI
SoftwareII
minershell: $
```

You can also execute commands with **at least one pipe** (e.g. `ls | wc`) as shown below, as well as taking any input on `ls` command.

```
Shell — minershell ◀ runshell.sh — 80x24
minershell: $ ls
DatabaseManagement      Resume                  WebsiteDevelopment
Graduation              SoftwareEngineering
RealEstate              Videos
minershell: $ cd Resume
minershell: $ ls
CoverLetter.docx        InterviewSheet.docx    Resume.pdf
CoverLetter.pdf         Resume.docx
minershell: $ ls Resume.docx | wc
      1      1     12
minershell: $ ls | wc
      5      5     76
minershell: $
```

Hurdles faced while implementing:

Out of all the labs regarding the shell, I personally feel like piping was the most difficult to implement into my existing shell due to the fact that it is such a hassle to execute, in my opinion. Adding it into my existing function 'execute' was nearly enough, few errors came up, but at the end, all was resolved and implemented the right way.

References:

<https://people.cs.rutgers.edu/~pxk/416/notes/c-tutorials/pipe.html>

[https://en.wikipedia.org/wiki/Pipeline_\(Unix\)](https://en.wikipedia.org/wiki/Pipeline_(Unix))