

Assignment 1 Overview and Implementations

main

- prompts
- parses line
- calls handler
- loops forever

↳ calls execute

↳ Calls executePipes

handler

- job: takes array of strings in and formats into a temp array that execvp can call
 - ↳ bc execvp doesn't want anything but commands and args
- recognizes input/output files
- calls execute appropriately for given special characters
- takes start & end index of args to read
 - implemented this way to handle ";"

Execute

- job: takes in a formatted array of strings containing ONLY command & args
 - takes in input & output files
 - executes the command

executePipes

- job: executes a formatted array of strings containing multiple commands given index of commands

Assignment 1 Design Doc

main

while(1)

 prompt()

 getLine()

 getSize()

 handler(args, 0, size)

handler(*args, start, end)

 **temp, counter, input, output

 if(args[start] == 0)

 exit()

 for(i = start → end) // loop through args and store appropriate

 if(args[i] == ">")

 output = args[++i]

 elif(== "<")

 input = args[++i]

 elif(== ">>")

 output = args[++i]

 appendFlag = ON

 elif(== " ; ")

 break

 elif (other special characters)

 handle appropriately

 else // not special or file so command or arg

 temp[counter] = args[i]

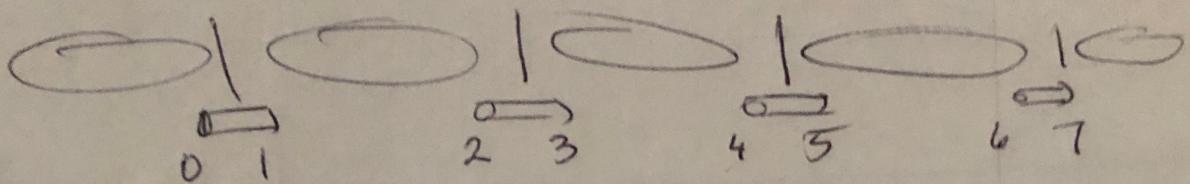
 temp[+counter] = NULL

 execute(temp, input, output, appendFlag)

 if (i < end)

 handler(args, i+1, end)

```
execute(*args, input, output, appendFlag)
    int fdin, fdout
    pid_t pid = fork();
    if (pid < 0) error
    elif (pid > 0) waitpid(pid) //parent process
    else //child process
        if (input != NULL)
            fdin = open(input)
            : //error handle
        if (output != NULL) ← if (appendFlag)
            fdout = open(output)
            : //error handle
        execv(args[0], args)
        : //error handle
```



- Pipes

- given an array of commands and args, how can you output execute these commands piped to the subsequent command
- array = [command, "args", NULL, input, output, pipe]
- Pipe = 4
 - 4 pipes detected
 - 5 commands to exec

```

execPipes(args, Input, Output, NumPipes, I
int pipes[NumPipes+2]                                CMDIndex]
                                         NumPipes×2
int j=0;
for (int i=0; i<#Pipes; i++)
    pipe(pipes + (2×i)); 0, 2, 4
if (fork() == 0)
    close(0);
    fdin = open(input);
    dup(fdin);
    close(1);
    dup(p[i]);
    for (i=0 → #Pipes)
        close(p[i]); execute (args[0], args)
else
    if (fork() == 0)
        for (0 → NumPipes+1 & i ≠ 0) {
            if (fork() == 0)
                close(0); dup(p[i]); close(1); dup(p[i+2]);
                closePipes(p); exec(args[CMDIndex[j]] & args)
            else
                waitpid

```