COMP 304 FALL 2022 – PROJECT 1

DEADLINE: NOVEMBER 17, 2022

ATAKAN ÖZKAN 76277 | DEPARTMENT OF COMPUTER ENGINEERING, KOC UNIVERSITY

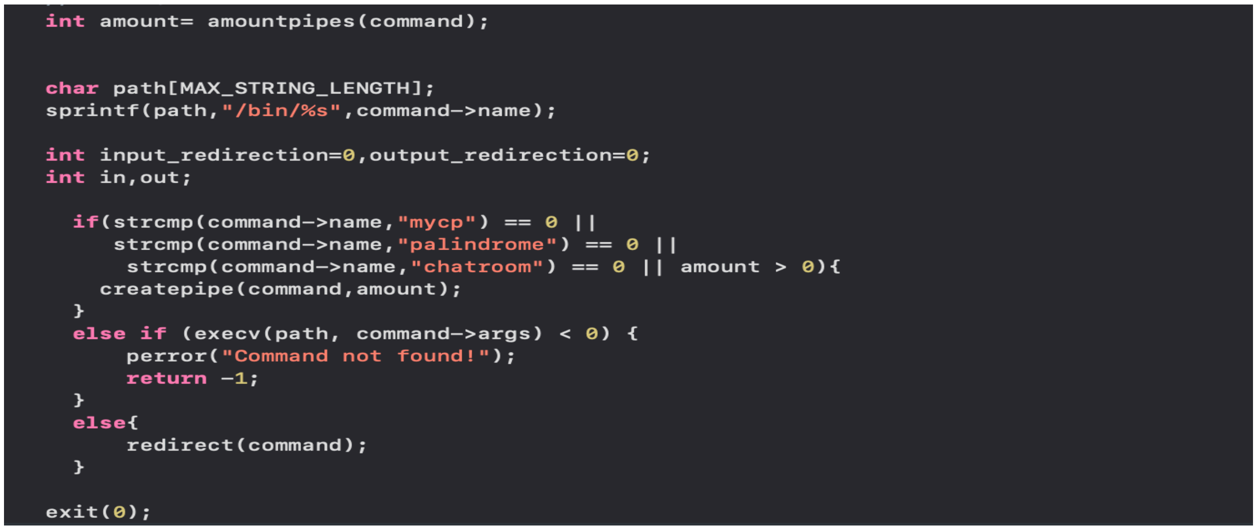
BUĞRAHAN YAMAN 76070 | DEPARTMENT OF COMPUTER ENGINEERING, KOC UNIVERSITY

1) INTRODUCTION

In this project, we worked on Unix-style operating system shell called Shellax. The assignment consists of different parts where we build our Unix command or upgrade The functionality of our Unix-style shell.

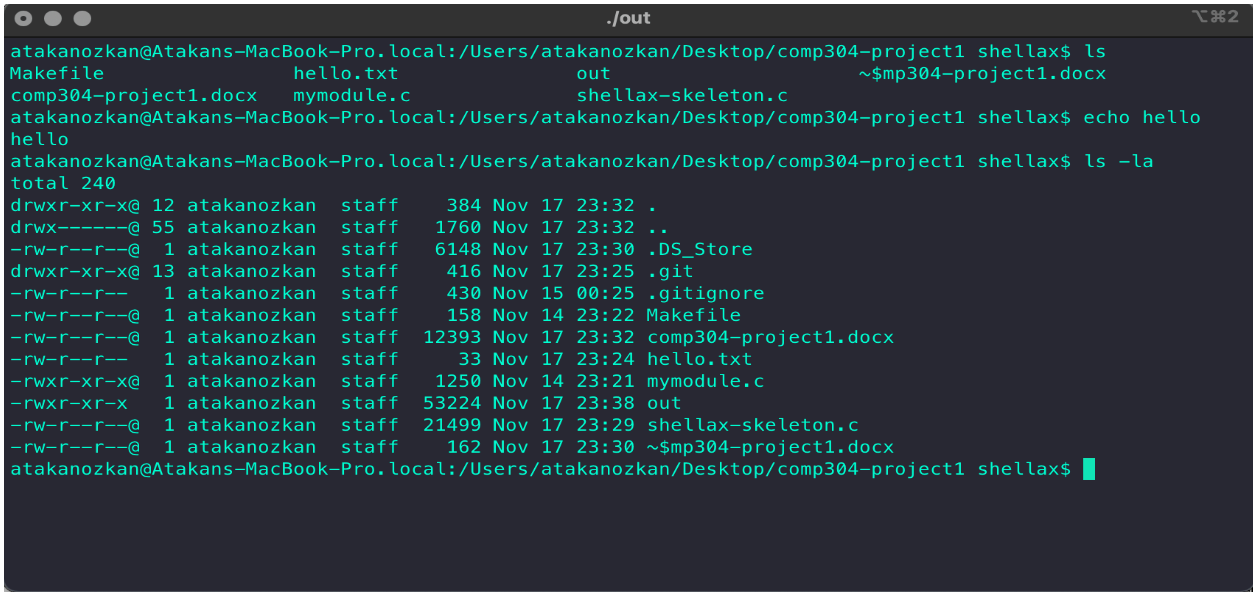
2) SYSTEM CALL WITH EXECV()

We implemented our execv() style system call to execute programs. The output of launching the programs immediately returns the command line prompt.



```
int amount= amountpipes(command);


char path[MAX_STRING_LENGTH];
sprintf(path,"/bin/%s",command->name);

int input_redirection=0,output_redirection=0;
int in,out;

  if(strcmp(command->name,"mycp") == 0 ||
     strcmp(command->name,"palindrome") == 0 ||
      strcmp(command->name,"chatroom") == 0 || amount > 0){
    createpipe(command,amount);
  }
  else if (execv(path, command->args) < 0) {
      perror("Command not found!");
      return -1;
  }
  else{
      redirect(command);
  }

exit(0);
```

As seen in the figure we execute our programs if they don't have any pipes or specific command. We execute the bult-in commands with execv() function.

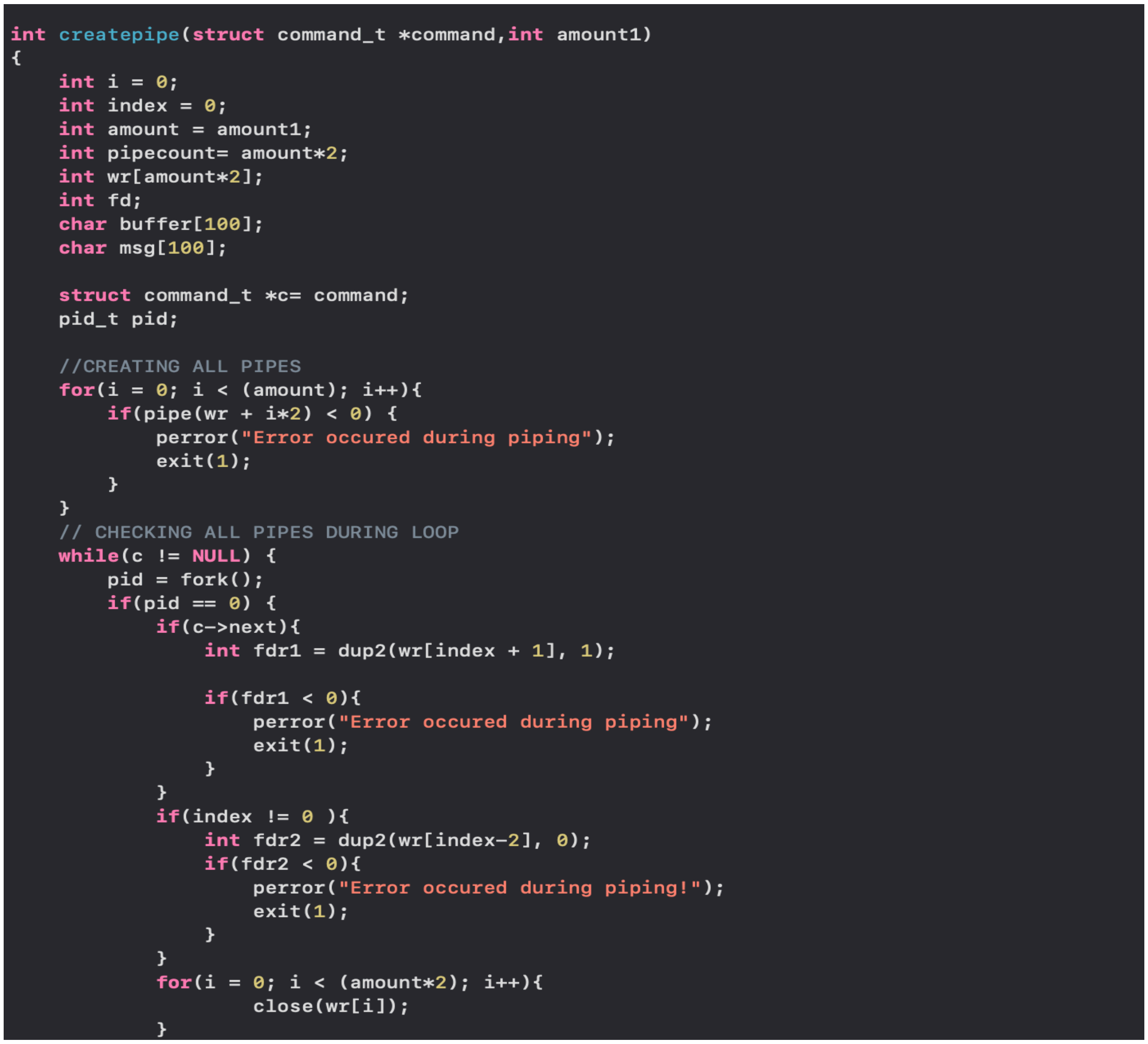


```
atakanozkan@Atakans-MacBook-Pro.local:/Users/atakanozkan/Desktop/comp304-project1 shellax$ ls
Makefile                hello.txt               out                     ~$mp304-project1.docx
comp304-project1.docx   mymodule.c              shellax-skeleton.c
atakanozkan@Atakans-MacBook-Pro.local:/Users/atakanozkan/Desktop/comp304-project1 shellax$ echo hello
hello
atakanozkan@Atakans-MacBook-Pro.local:/Users/atakanozkan/Desktop/comp304-project1 shellax$ ls -la
total 240
drwxr-xr-x@ 12 atakanozkan  staff    384 Nov 17 23:32 .
drwx------@ 55 atakanozkan  staff   1760 Nov 17 23:32 ..
-rw-r--r--@  1 atakanozkan  staff   6148 Nov 17 23:30 .DS_Store
drwxr-xr-x@ 13 atakanozkan  staff    416 Nov 17 23:25 .git
-rw-r--r--   1 atakanozkan  staff    430 Nov 15 00:25 .gitignore
-rw-r--r--@  1 atakanozkan  staff    158 Nov 14 23:22 Makefile
-rw-r--r--@  1 atakanozkan  staff  12393 Nov 17 23:32 comp304-project1.docx
-rw-r--r--   1 atakanozkan  staff     33 Nov 17 23:24 hello.txt
-rwxr-xr-x@  1 atakanozkan  staff   1250 Nov 14 23:21 mymodule.c
-rwxr-xr-x   1 atakanozkan  staff  53224 Nov 17 23:38 out
-rw-r--r--@  1 atakanozkan  staff  21499 Nov 17 23:29 shellax-skeleton.c
-rw-r--r--@  1 atakanozkan  staff    162 Nov 17 23:30 ~$mp304-project1.docx
atakanozkan@Atakans-MacBook-Pro.local:/Users/atakanozkan/Desktop/comp304-project1 shellax$
```

This is the output after we execute the matching bult-in command.

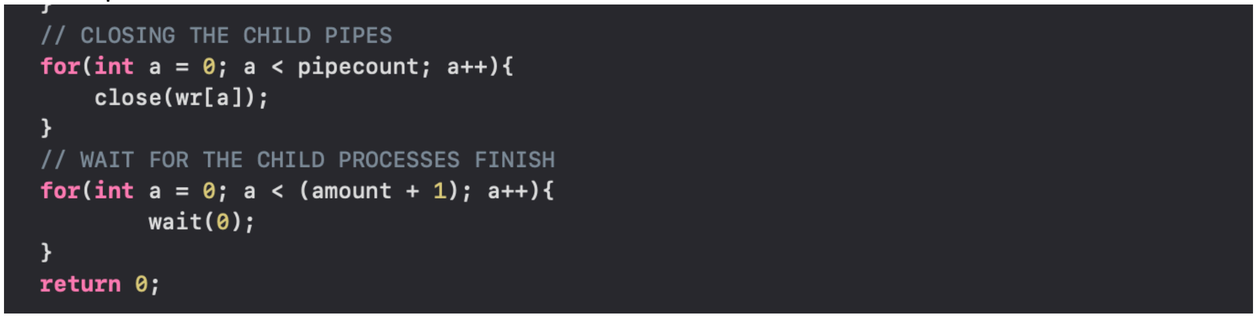
3) HANDLING PIPING FOR SHELLAX

The input sometimes contains the pipes. To enable the piping, we created needed children to connect the other pipes. Supporting more than two pipes, allow us to handle the chain of pipes.



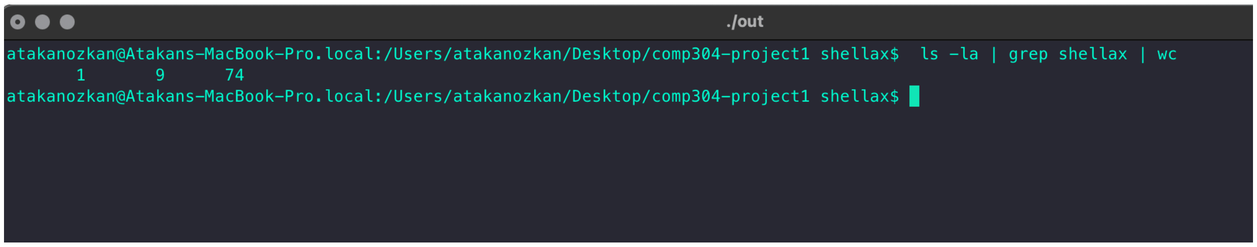
```c
int createpipe(struct command_t *command,int amount1)
{
    int i = 0;
    int index = 0;
    int amount = amount1;
    int pipecount= amount*2;
    int wr[amount*2];
    int fd;
    char buffer[100];
    char msg[100];

    struct command_t *c= command;
    pid_t pid;

    //CREATING ALL PIPES
    for(i = 0; i < (amount); i++){
        if(pipe(wr + i*2) < 0) {
            perror("Error occured during piping");
            exit(1);
        }
    }
    // CHECKING ALL PIPES DURING LOOP
    while(c != NULL) {
        pid = fork();
        if(pid == 0) {
            if(c->next){
                int fdr1 = dup2(wr[index + 1], 1);

                if(fdr1 < 0){
                    perror("Error occured during piping");
                    exit(1);
                }
            }
            if(index != 0 ){
                int fdr2 = dup2(wr[index-2], 0);
                if(fdr2 < 0){
                    perror("Error occured during piping!");
                    exit(1);
                }
            }
            for(i = 0; i < (amount*2); i++){
                    close(wr[i]);
            }
```

In the createpipe() function we handle all pipes. Firstly we create the pipes then we put them into loop where we connect them in a row.



```c
  // CLOSING THE CHILD PIPES
  for(int a = 0; a < pipecount; a++){
      close(wr[a]);
  }
  // WAIT FOR THE CHILD PROCESSES FINISH
  for(int a = 0; a < (amount + 1); a++){
          wait(0);
  }
  return 0;
```

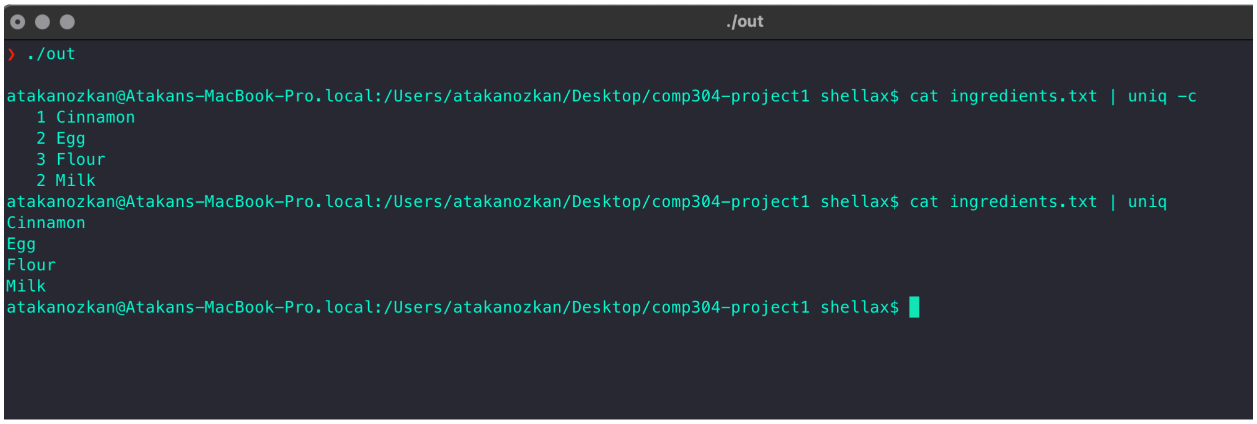
After we are done with all pipes and the children, we simply terminate them.



```
atakanozkan@Atakans-MacBook-Pro.local:/Users/atakanozkan/Desktop/comp304-project1 shellax$  ls -la | grep shellax | wc
       1       9      74
atakanozkan@Atakans-MacBook-Pro.local:/Users/atakanozkan/Desktop/comp304-project1 shellax$
```

4) UNIQ COMMAND

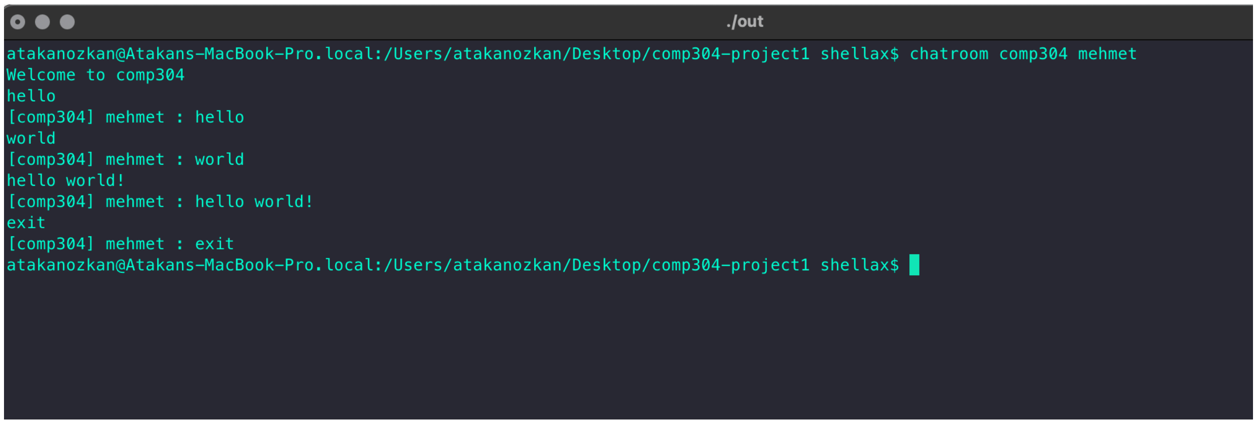
Using uniq command in the Shellax allows us to find unique words. If -c parameter given then, it will print the count of those words.



```
❯ ./out

atakanozkan@Atakans-MacBook-Pro.local:/Users/atakanozkan/Desktop/comp304-project1 shellax$ cat ingredients.txt | uniq -c
   1 Cinnamon
   2 Egg
   3 Flour
   2 Milk
atakanozkan@Atakans-MacBook-Pro.local:/Users/atakanozkan/Desktop/comp304-project1 shellax$ cat ingredients.txt | uniq
Cinnamon
Egg
Flour
Milk
atakanozkan@Atakans-MacBook-Pro.local:/Users/atakanozkan/Desktop/comp304-project1 shellax$
```

5) CHATROOM

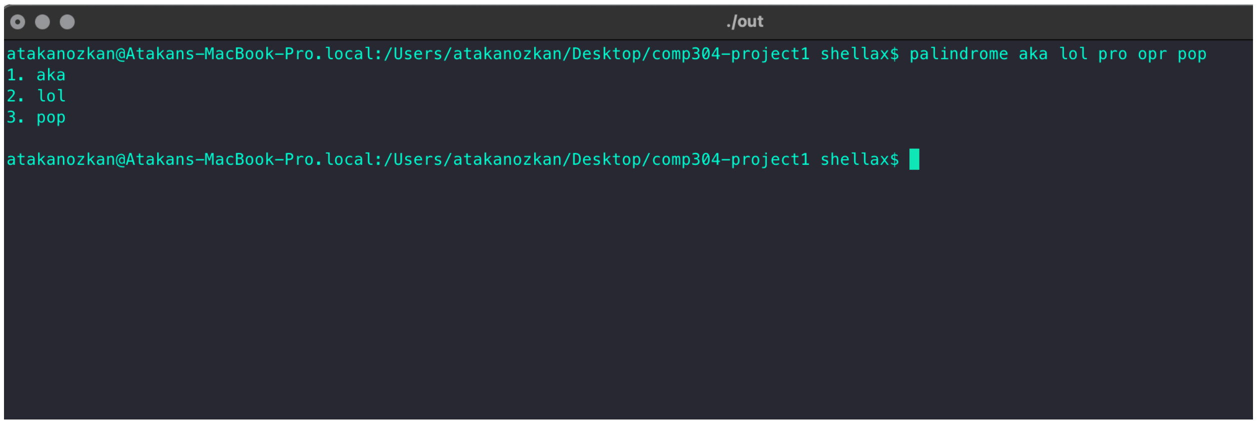
This is the chatroom of comp304 folder. The user just writes in the named pipe after we write to all named pipes under the chatroom folder.



```
atakanozkan@Atakans-MacBook-Pro.local:/Users/atakanozkan/Desktop/comp304-project1 shellax$ chatroom comp304 mehmet
Welcome to comp304
hello
[comp304] mehmet : hello
world
[comp304] mehmet : world
hello world!
[comp304] mehmet : hello world!
exit
[comp304] mehmet : exit
atakanozkan@Atakans-MacBook-Pro.local:/Users/atakanozkan/Desktop/comp304-project1 shellax$
```

6) PALINDROME COMMAND

This is the command that we implemented. This command basically finds the palindrome words in the arguments.



```
atakanozkan@Atakans-MacBook-Pro.local:/Users/atakanozkan/Desktop/comp304-project1 shellax$ palindrome aka lol pro opr pop
1. aka
2. lol
3. pop

atakanozkan@Atakans-MacBook-Pro.local:/Users/atakanozkan/Desktop/comp304-project1 shellax$ 
```