**CS 307 - HOMEWORK 1** **Melisa Yılmaz 26777**

**man grep | grep “-A” 1 “\\-U” > output.txt**

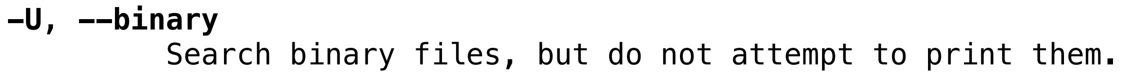
Grep command searches a file for particular pattern of characters that is given by the user and displays all lines that contain that chosen pattern.

-A n: is used for printing searched line and n lines after the result.
-U: is for searching binary files, but do not attempt to print them.

I picked this command which is the left side of the pipe because I want to show the user, reference manual of the grep command.

Also, at the right side of the pipe, I chose the -A 1 to see 1 line after the searched character which is “-U”.

As a result of that, in the terminal, there is seen the manual page of the grep which is shown as the searching of the character “-U" and representing 1 lines after that searched character.



```
-U, --binary
        Search binary files, but do not attempt to print them.
```

## Hierarchy of the Program

In my pipeSim.c file, I tried to create a main part, parent process, using int main(). My aim was to create two child for the parent process and try to achieve the communication of this child with pipe().

Because of that, first of all, in main part, I created pipe as int fd[2]. fd[0] for reading, fd[1] for writing.

And then, I used fork() to create first child. This first was the left-side of the pipe, so man command part. I closed the fd[0] because I did not use it in this part. The child process of man dups its standard out to the pipe, I mean that using dup2(fd[1], STDOUT_FILENO), I achieved that making output go to the pipe and this output was achieved using execvp to run my command.

Secondly, I created second child which includes grep command. I closed fd[1] because I did not use it. The child process of grep dups its fd[0] to the stdin as using dup2(fd[0], STDIN_FILENO) which is getting input from the pipe and I created int new_fd to copy stdout to the output.txt so the result of these commands can appears in that txt file.

Finally, in my main/parent process, I closed the fd[0] and fd[1] and I used waitpid since I have two processes, I needed to clarify which processes should parent waits. At the end my program returns 0.