Zeynep Türkmen - 29541

Programming Assignment 1

```
man touch | grep -A 1 -m 1 -e "-t" > output.txt
```

touch command allows us to create new files and to edit the access/modification timestamps of a file.

```
touch something
```

Above command creates a new file called “something” if it doesn’t already exist. If it *does* exist, it modifies the access and modification timestamps of that file to the *current time.*

-t flag helps us pick the timestamps we wish as shown below:

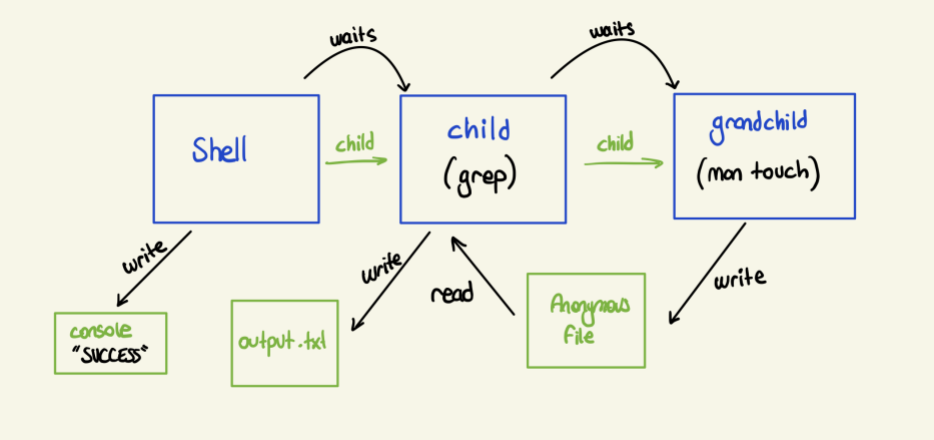
```
touch -t 192310291000 something.txt
```

This will change the access and modification times of something.txt as 23 October 1923 10 am.

I picked this option/command because I like playing with the terminal and I always use *touch* to create txt files on macOS (it’s the simplest way to create .txt). So, I basically wanted to discover some new options for it and -t seemed like the best one. I can open a file and simply make it look like I didn’t *touch* it. I also can edit a file and make it look like it was edited before I was born, which is cool.

The program hierarchy:

I used 2 forks, 2 execvp and a pipe for my assignment. The process diagram is something like this:



How it works?

The main shell program forks first, and then the child calls another fork function.
The grandchildren change the standard output file descriptor as the anonymous file of the pipe (using dup2), then it executes the **'man touch'** command. This helps writing the result of the command into the anonymous file.

Next, the child starts playing. It first waits for the grandchildren to finish their execution. After making sure, it changes the standard input file descriptor again using dup2, to the anonymous file (which includes the result of the man command). It also opens a file called "output.txt" and changes the standard output file descriptor as output. This will allow the child to read from grandchild result and write into the output file. After that it executes **"grep -A 1 -m 1 -e "-t"".**
-m 1 => tells grep to get only one occurrence
-A 1 => manages how many lines should be displayed after a match occurs (1 is enough for -t)
-e => tells the shell that -t is not an option, but a search pattern (Note that: I also tried the \ character which also works but I wanted to use something from the manual)
Now the result is all inside output.txt.

Finally, the parent (i.e., the shell) waits for the child to finish executing and prints a message indicating that the execution is complete.