Project 1 Report

Comp 304

KUSIS ID:49928 /PARTNER-NAME: Berke Ağababaoğlu

KUSIS ID:49752 /PARTNER-NAME: Berkan Hızıroğlu

    In this project there are four parts. In the first part we use “execvp” system call to call built in functionalities of linux machine. We could have used “execv” but it would only invoke the functions that are in /bin directory, that is why we stick with the “execvp”. Our function, forks a child, child does all the job, parent waits for the child, and then continue. For part two, part a, we checked if there is a redirection and if there is, we redirect output to a file using dup2. **For part 2**, part b, if user invokes script our function creates a file and sets the script value to one and then whenever user inputs our functions writes it in to the file.

**For part 3**, in bookmark we first parsed the input string if the first argument is “bookmark”. Then, we created the required file and stored the key-value pairs to that file.

To execute the bookmarked command, we first open the file. Read it line by line to find the previously stored key and it’s corresponding key. Then, we call execvp() to execute it.

We took the linked-list implementation from an online source.

**For part 3**, “wforecast”, we use crontabs and curl as suggested. Whenever user inputs “wforecast”, we create an .sh file in home folder for crontab to execute and edit the crontab file in /var/spool/cron/crontabs/username to execute our command every morning 9am. For part 3, our command, we implement a command called “commentout” that takes a file as input and then returns the commented-out version of the text file with the same name.

**For part 4,** we pretty much followed the exact same things in the link that is provided in the project manual. We make N number of forks where N is the length of the input string in a for loop. The first child takes the stored array and by p=0.5 probability, changes the first character to its successor in the alphabet. It then puts the message in the POSIX queue created. After that, it unlinks itself from the queue and calls exit(0). At the next iteration of the for loop, we create the next child. The children who are not the first one, first calls receive to get the stored message in the queue. While doing all these, the parent waits for all the children to finish. After receiving that message, each children changes the string’s corresponding character according to their index with p=0.5 probability and sends the message to the queue again.