CSE344 - HW1 Report

Abdullah Çelik

March 2022

1 Overview

- Firstly, all requirements were achieved.
- The checkArgument function checks if the arguments are valid or not. If checkArgument function encounters a non-valid input while doing this, it edits its own global variable. printArgumentError prints a message to stderr according to this global variable. Here, I used the design of syscalls regulating errno in case of an error and the perror function to print output according to errno.
- The parseRule function parses the rules given as arguments (RULE.h, RULE.c) and adds them into a generic vector (the generic vector class VECTOR.h, VECTOR.c) and then returns this vector.
- It is opened for reading and writing with given input file with read syscall. Before editing the file, it tries to lock the file using fcntl syscall as write lock(exclusive). This is to prevent multiple instances from trying to edit the same file at the same time. Otherwise, the input will be corrupted. The reason why I locked file as write lock(exclusive lock) is that I will be reading first and then writing on the same file. Because when I locked file as write lock, the file is closed for both reading and writing. Then temp file is created with mkstemp syscall and unlinked with unlink syscall. The reason for this is that the space used by the temp file can be reused if the temp file is closed and then its association with the name is lost. Each word read from the input file is compared with the rules. If they do not match, the word read from the input file is written to the temp file, if they match, the new word given in the rule is written. This ends the reading process. Then the inputfile is cleared with truncate syscall, offsets are set to 0 with lseek syscall and whatever is in the temp file is copied to the input file. Read and write syscalls are used in copying process too. The input file is unlocked with fcntl syscall. The input file and temp file are close with close syscall.

2 Error Handling

- SYSCALL
  Description: The case where syscalls returns an error
  Action: Prints error message using perror and terminates program

- Allocating memory
  Description: The case where there is not enough space when allocating memory
  Action: Prints error message to stderr and terminates program

- Number of arguments
  Description: The case where user enters more or less arguments
  Action: Prints program usage to stderr and terminates program

- Rule usage
  Description: The case where user doesn't enter the rule expected format, misuse of slashes
  Action: Prints rule usage to stderr and terminates program

- The first part which contains regex of the rule's characters
  Description: The case where the user doesn't comply with the contract. The first part which contains regex of the rule can contain only letters, digits and "ˆ $ [ ] *" characters
  Description: The case where the first part which contains regex of the rule can't contain any letters or digits
  Action: Prints useful message and terminates program

- The second part of the rule
  Description: The case where the user doesn't comply with the contract. The second part of the rule can contain only letters, digits
  Action: Prints useful message and terminates program

- Usage of ˆ character
  Description: Incorrect use of the ĉharacter. ĉharacter can only be found at the beginning
  Action: Prints useful message and terminates program

- Usage of $ character
  Description: Incorrect use of the $ character. $ character can only be found at the end
  Action: Prints useful message and terminates program

- Usage of [ ] characters
  Description: Incorrect use of the [ ] characters
  Action: Prints useful message and terminates program

- Usage of * characters
  Description: Incorrect use of the * character. Letter, digit or properly

used brackets can come before * character. * character cannot follow * character
Action: Prints useful message and terminates program

3 Compile and Run

- make → Compiles the whole program
  Type make in the file contains the makefile
- make clean → Cleans all objects files