

## **K24: Προγραμματισμός Συστήματος - 1η Εργασία**

### **Εαρινό Εξάμηνο 2023**

**Παναγιώτης Κοντοειδής**  
**1115201900266**

**Το tar.gz αρχείο περιλαμβάνει τα εξής αρχεία:**

- **Makefile**
- **libs.h**
- **shell.cpp**
- **parser.cpp**
- **redirections.cpp**
- **pipes.cpp**
- **alias.cpp**
- **history.cpp**
- **syspro2023\_hw1\_completion\_report\_1900266.docx**

**Για την εκτέλεση του κελύφους καλούνται οι εντολές:**

-Make  
-./mysh

**Για το κλείσιμο του κελύφους εκτελείται η εντολή:**

-exit

### **Σχεδιαστικές επιλογές**

#### **Parsing:**

Η εντολή που θα δοθεί περνάει από μία σειρά από parser που εκτελούν διαφορετική λειτουργία. Αρχικά, η εντολή αποθηκεύεται σε ένα `vector<char*>`. Ο πρώτος parser, `read_command`, χωρίζει σε tokens το command με delimiter " ". Ακολουθεί ο parser που εντοπίζει στο ήδη υπάρχων command τους χαρακτήρες των pipes και των redirections, `redirections_pipes_in_cmd`. Ο parser των redirections μπορεί να εντοπίσει μέσα σε tokens τα σύμβολα ανακατεύθυνσης ">", "<", και το διαχωρίζει από το token όμως για να

εντοπιστεί το σύμβολο της ανακατεύθυνσης προσθήκης " >>" θα πρέπει να προηγείται και ακολουθείται μετά από αυτό κενό, δηλαδή να έχει γίνει ήδη token από τον `read_command`. Για παράδειγμα δεν εντοπίζεται το `a>>b` αλλά μόνο το `a >> b`. Στην συνέχεια ακολουθεί ο parser των μεταβλητών περιβάλλοντος, `env_variables`, όπου και αντικαθιστά στο `command` την τιμή της μεταβλητής. Τελευταίος είναι ο parser για `aliases`, `check_for_aliases`, όπου και αντικαθιστά το `alias` με την τιμή που του έχει δοθεί. Ο parser αυτός καλεί αν χρειαστεί την συνάρτηση `split_command` η οποία κάνει την ίδια δουλειά με τον `read_command` και επιστρέφει την τιμή του `alias` χωρισμένη με `delimiter` το κενό.

### **Redirections:**

Υποστηρίζονται κανονικά οι μονές και οι διπλές ανακατευθύνσεις και η ανακατεύθυνση προσθήκης σε ήδη υπάρχον αρχείο. Μόλις εντοπιστεί σύμβολο ανακατεύθυνσης καλείται η `call_for_redirection` η οποία ανακατευθύνει ανάλογα το σύμβολο.

### **Pipes:**

Για την υλοποίηση των σωληνώσεων χρησιμοποιήθηκε κώδικας από τις διαφάνειες της διάλεξης 9 σελίδα 53. Υποστηρίζεται η παροχή μια σωλήνωσης και ο συνδυασμός της με ανακατευθύνσεις. Η διαδικασία παροχής σωλήνωσης ξεκινάει με τον εντοπισμό του συμβόλου σωλήνωσης όπου και καλείται η `call_for_pipes`.

### **Διαχείριση Alias:**

Τα `alias` αποθηκεύονται σε ένα `vector<char*, char*>`.

Για να δημιουργηθεί ένα `alias` εκτελείται η εντολή

```
createalias <alias-name> "<alias command>";
```

Σε περίπτωση που χρησιμοποιηθεί όνομα για `alias` το οποίο ήδη έχει καταχωρηθεί η εντολή που αντιστοιχεί σε αυτό γίνεται `overwrite` από την καινούργια.

Για να καταστραφεί ένα alias εκτελείται η εντολή

```
destroyalias <alias-name>;
```

Επίσης για χρηστικότητα δημιουργήθηκε η εντολή print όπου με την κλήση της τυπώνεται ο πίνακας των alias.

### **Διαχείριση Σημάτων:**

Για την διαχείριση σημάτων στο βασικό loop που εκτελείται το κέλυφος τα σήματα SIGINT και SIGTSTP αγνοούνται και όταν εκτελείται είναι να εκτελεστεί κάποια εντολή μετά την fork επαναφέρονται τα σήματα μέσα στο παιδί όσο περιμένει ο πατέρας.

### **Αποθήκευση Ιστορικού:**

Για να εκτυπωθεί το ιστορικό εκτελείται η εντολή

```
myHistory
```

Για να εκτελεστεί μια εντολή από το ιστορικό εκτελείται η εντολή

```
myHistory <number>
```

Το ιστορικό αποθηκεύει μέχρι 20 εντολές και επίσης μπορεί να γίνει αναδρομική κλήση εντολών ιστορικού πχ.

1. ls -l
2. myHistory 1
3. myHistory 2

Η κλήση της εντολής myHistory 3 θα καλέσει τελικά την εντολή ls -l.