

- Πρώτος Φλοιός (auebsh1)

Αρχικά, ο συγκεκριμένος φλοιός εκτελεί απλές μεμονωμένες εντολές, π.χ. `ls`, `pwd`, `date`, `cal` κτλ. Για να επιτευχθεί αυτό δημιουργούμε 6 μεταβλητές, όπου ο ρόλος τους είναι ο εξής:

- `static char line[1024]` : η εντολή που δίνει ο χρήστης από το πληκτρολόγιο,
- `int input` : η τιμή από την προηγούμενη εντολή (χρησιμοποιείται αργότερα για τη σωλήνωση),
- `int first` : η πρώτη λέξη-εντολή (π.χ. `ls`) της εντολής που έχει δώσει ο χρήστης (χρησιμοποιείται αργότερα για τη σωλήνωση),
- `int temp` : το μέγεθος του `line`,
- `int permission` : αν έχει την τιμή 0 σημαίνει ότι ο φλοιός δε δέχεται παραμέτρους,
- `char* cmd` : δείκτης που δείχνει στο `line` για να μπορέσουμε να επεξεργαστούμε την εντολή χωρίς να την αλλοιώσουμε.

Καλώντας τη συνάρτηση `run`, που δέχεται σαν ορίσματα όλες τις παραπάνω μεταβλητές, πλην της `line`. Εφόσον έχουμε `permission=0` καλούμε την `split`, η οποία με τη βοήθεια της συνάρτησης `skipspace` που αγνοεί τα κενά από το `cmd`, χωρίζει το `cmd` σε tokens τα οποία αποθηκεύονται στον πίνακα `args[]`. Με την προϋπόθεση ότι ο χρήστης έχει δώσει κάποια εντολή, καλείται η συνάρτηση `command`. Η συνάρτηση αυτή εκτελεί τις εντολές που έχουν δοθεί (στο συγκεκριμένο φλοιό εκτελεί τη μεμονωμένη εντολή που έχει δοθεί από το χρήστη).

Ο φλοιός μπορεί να τερματίζει όταν διαβάσει τον ειδικό χαρακτήρα EOF ή αλλιώς όταν ο χρήστης πληκτρολογεί CTRL-D.

- Δεύτερος Φλοιός (auebsh2)

Ακολουθείται η ίδια διαδικασία με τον πρώτο φλοιό, με τις ίδιες μεταβλητές. Η διαφορά τους είναι ότι στη *run* γίνεται έλεγχος για τις ανακατευθύνσεις εισόδου/εξόδου (< / >) και ότι ανοίγει ή/και δημιουργείται ένα αρχείο. Οι μεταβλητές που χρησιμοποιούνται επιπλέον είναι οι εξής:

- *char rfile[20]*: αποθηκεύει το όνομα του αρχείου εισόδου (μετά το '<'),
- *char wfile[20]*: αποθηκεύει το όνομα του αρχείου εξόδου (μετά το '>'),
- *int textin*: χρησιμοποιείται για το άνοιγμα του αρχείου και για την υποβολή των δικαιωμάτων του,
- *int textout*: χρησιμοποιείται για το άνοιγμα του αρχείου και για την υποβολή των δικαιωμάτων του.

- Τρίτος Φλοιός (auebsh3)

Ακολουθείται η ίδια διαδικασία με τους δύο προηγούμενους φλοιούς. Η διαφορά τους είναι ότι η μεταβλητή *permission* παίρνει την τιμή 1 και ότι ο φλοιός μπορεί να εκτελέσει εντολές με παραμέτρους (π.χ. *sort -u*, *ls -l*).

- Τέταρτος Φλοιός (auebsh4)

Ακολουθείται η ίδια διαδικασία με τους τρεις προηγούμενους φλοιούς. Η διαφορά τους είναι ότι εκτελούνται εντολές με σωλήνωση και χρησιμοποιούνται τρεις επιπλέον μεταβλητές:

- *int first*: έχει την τιμή ένα όταν αναφερόμαστε στην πρώτη εντολή της σωλήνωσης (όταν δεν έχουμε είσοδο από το προηγούμενο σωλήνα),
- *int last*: έχει την τιμή ένα όταν αναφερόμαστε στην τελευταία εντολή της σωλήνωσης (όταν δεν έχουμε είσοδο από το προηγούμενο σωλήνα),

- char* first pipe : ο δείκτης αυτός δείχνει την εντολή μετά την πρώτη σωλήνωση (' | ') και χρησιμοποιείται για να ελέγξουμε αν υπάρχουν πάνω από 2 σωληνώσεις.

Για παράδειγμα εάν ο χρήστης πληκτρολογήσει την εντολή `ls -l /home/csuser/Downloads | sort -u > listing.txt` τότε το file descriptor θα πάρει τις τιμές:

- `fd1 = command(0, 1, 0, "\0", "\0")`, με `args[0] = "ls"`, `args[1] = "-l"` και `args[2] = "/home/csuser/Downloads"`
- `fd2 = command(fd1, 0, 1, "\0", "listing.txt")`, με `args[0] = "sort"`, `args[1] = "-u"` και `args[2] = "listing.txt"`

Επομένως παρατηρούμε ότι το input παίρνει στο `fd2` την τιμή του `fd1` όπως εξηγήσαμε στο `auebsh1`.

- Πέμπτος Φλοιός (auebsh5)

Ακολουθείται η ίδια διαδικασία με τους τέσσερις προηγούμενους φλοιούς. Η διαφορά τους είναι ότι εκτελούνται εντολές με σωλήνωση χωρίς περιορισμό στο πλήθος τους.

Για παράδειγμα εάν ο χρήστης πληκτρολογήσει την εντολή `ls -l /home/csuser/Downloads | sort -u | wc -l > count.txt` τότε το file descriptor θα πάρει τις τιμές:

- `fd1 = command(0, 1, 0, "\0", "\0")`, με `args[0] = "ls"`, `args[1] = "-l"` και `args[2] = "/home/csuser/Downloads"`
- `fd2 = command(fd1, 0, 0, "\0", "\0")`, με `args[0] = "sort"`, `args[1] = "-u"`
- `fd3 = command(fd2, 0, 1, "\0", "count.txt")`, με `args[0] = "wc"`, `args[1] = "-l"`, `args[2] = "count.txt"`