

1τη ΠΡΟΓΡΑΜΜΑΤΙΣΤΙΚΗ ΕΡΓΑΣΙΑ ΣΤΑ ΛΕΙΤΟΥΡΓΙΚΑ ΣΥΣΤΗΜΑΤΑ

ΜΑΥΡΑΓΑΝΗΣ ΒΑΣΙΛΕΙΟΣ 3160091

ΚΟΚΚΟΚΥΡΗΣ ΣΤΥΛΙΑΝΟΣ 3160063

ΤΟΣΚΟΛΛΑΡΙ ΡΟΝΑΛΝΤ 3160244

aeubsh1: Αρχικά χρησιμοποιούμε την μέθοδο `getCommand()` η οποία παίρνει την εντολή που πληκτρολογήσαμε στην γραμμή εντολών και την τοποθετεί σε ένα πίνακα χαρακτήρων χαρακτήρα-χαρακτήρα. Μετά χρησιμοποιούμε την `parseCommand()` η οποία δέχεται σαν όρισμα τον πίνακα χαρακτήρων που επιστρέφει η `getCommand()` για να διαχωρίσουμε τις εντολές από τις παραμέτρους. Αφού διαχωρίσουμε εντολές και παραμέτρους αποθηκεύοντας τα στον πίνακα `args`, χρησιμοποιούμε την `execute()` η οποία δέχεται ως όρισμα τον `args` για να εκτελέσουμε την εντολή που πληκτρολογήσαμε δημιουργώντας μια θυγατρική διεργασία με την `fork()`.

aeubsh2: Ο δεύτερος φλοιός χρησιμοποιεί στην αρχή τις μεθόδους `getCommand()` και `parseCommand()` όπως το `aeubsh1`. Έπειτα, ο πίνακας `args` που περιέχει διαχωρισμένες τις εντολές και παραμέτρους δίνεται ως είσοδο στην μέθοδο `redirectAndExec()` η οποία αντικαθιστά στον πίνακα τα σύμβολα της ανακατεύθυνσης με `null` ώστε να μπορεί να αναγνωριστεί η εντολή. Εφ'όσον δεν χρησιμοποιούμε `pipes` στην `redirection` η επομένη εντολή που εκτελεί η `redirectAndExec()` είναι η `execute()` που εκτελεί την ανακατεύθυνση αφού δημιουργήσει την θυγατρική διεργασία μέσα στην οποία εκτελούμε την μέθοδο `redirect()` η οποία δημιουργεί τους κατάλληλους `file descriptors` για την ανακατεύθυνση.

aeubsh3: Ο τρίτος φλοιός είναι δομικά ίδιος με τον δεύτερο μόνο που πριν την `redirect` προσθέσαμε μια `while` η οποία αντικαθιστά την παράμετρο (-) με το `null` ώστε να μπορεί να αναγνωριστεί η εντολή και στην συνέχεια ακολουθούμε τον ίδιο τρόπο όπως και στο προηγούμενο ερώτημα.

aeubsh4: Στον προτελευταίο φλοιό χρησιμοποιούμε τις `getCommand()` και `parseCommand()` όπως στους προηγούμενους. Στην συνέχεια εκτελούμε την μέθοδο `exec_pipe()` η οποία υλοποιεί την `pipe`. Η `exec_pipe()` παίρνει ως παράμετρο τον πίνακα `args` ο οποίος όπως και πριν περιέχει διαχωρισμένες τις εντολές και παραμέτρους που δώσαμε στην γραμμή εντολών. Αρχικά στο εσωτερικό της μεθόδου εκχωρούμε δυναμικά χώρο για τις δύο εντολές χρησιμοποιώντας δύο πίνακες χαρακτήρων τους `args1` και `args2` και στην συνέχεια τις βρίσκουμε και τις τοποθετούμε στους πίνακες αντίστοιχα και μετά ελέγχουμε τους `args1, args2` αν υπάρχει ανακατεύθυνση στις εντολές και αποθηκεύουμε στις μεταβλητές `inFile, outFile` 1,2 τα ονόματα των αρχείων στα οποία θα γίνει. Έπειτα, αν υπάρχει `pipe` μεταξύ των 2 εντολών δημιουργούμε τους `file descriptors` και για κάθε εντολή 1) δημιουργούμε με την `fork()` την θυγατρική διεργασία 2) `dup` τους `file descriptors` 3) χρησιμοποιούμε την `redirect()` ώστε σε περίπτωση που έχουμε ανακατεύθυνση να ανοίξουμε `file descriptors` για να διαβάσουμε ή να γράψουμε 4) και στο τέλος εκτελούμε με την `execvp` την κάθε εντολή.

aeubsh5: Στον τελευταίο φλοιό αρχικά εκτελούμε τις `getCommand()` και `parseCommand`. Ο πίνακας `args` που περιέχει διαχωρισμένες τις εντολές και τις παραμέτρους δίνεται ως είσοδο στην `redirectAndExec` η οποία αντικαθιστά στον πίνακα τα σύμβολα της ανακατεύθυνσης με `null` ώστε να μπορεί να

αναγνωριστεί η εντολή. Αφού υλοποιούμε `pipes` θα εκτελεστεί η μέθοδος `exec_pipe2()` στην οποία κρατάμε έναν `buffer` με τα δεδομένα που προκύπτουν από την εκτέλεση των πολλαπλών `pipes` και στην συνέχεια αφού δημιουργήσουμε τους `file descriptors` χρησιμοποιούμε τις συναρτήσεις `read()`, `write()` για να γράψουμε τα δεδομένα στα αρχεία.