Λειτουργικά Συστήματα-Άσκηση 1η

Χαράλαμπος Κανελλόπουλος (4994)

Ερώτημα 1ο //

Στο 1ο ερώτημα μας ζητήθηκε να αναπτύξουμε ένα shell script σε C όπου ο χρήστης θα δίνει σαν είσοδο ένα πρόγραμμα. Ο σχεδιασμός του script εμπεριέχει την υλοποίηση αρχικά ενός command line όπου έχουμε το dollar sign σαν αρχή του τερματικού και έναν έλεγχο με μία επανάληψη για το πότε τερματίζετε το script. Χρησιμοποιώντας έπειτα την fork() ελέγχουμε πότε η τιμή της θα ισούται με το 0 όπου γνωρίζουμε από το documentation της fork() ότι επιστρέφει την διεργασία παιδί. Επίσης υπάρχει και μήνυμα σε περίπτωση λάθους παρόμοιο με το τερματικό του UNIX με την χρήση της fputs για να χρησιμοποιήσουμε ως έξοδο την stderr και όχι την stdout όπως διευκρινίστηκε στην εκφώνηση.

Ερώτημα 2ο //

Στο 2ο ερώτημα μας ζητήθηκε να αναπτύξουμε ένα shell script σε C όπου ο χρήστης θα δίνει σαν είσοδο ένα πρόγραμμα με πολλαπλές παραμέτρους. Ο σχεδιασμός του script εμπεριέχει την υλοποίηση αρχικά ενός command line όπου έχουμε το dollar sign σαν αρχή του τερματικού και έναν έλεγχο με μία επανάληψη για το πότε τερματίζετε το script.Έπειτα θα πρέπει να γίνεται έλεγχος
για το πότε ο χρήστης χρησιμοποιεί το κενό(« «) και θα γίνεται αποθήκευση των παραμέτρων αυτών σε έναν μονοδιάστατο πίνακα. Μέσω της fork() δημιουργείτε μία διεργασία παιδί, αντίγραφο της αρχικής για την εκτέλεση του προγράμματος.
Για την εκτέλεση των παραμέτρων χρησιμοποίησα την execvp() σε αυτό το ερώτημα επειδή χρησιμοποιώ πίνακα για την αποθήκευση τους.
Στο τέλος για την επιστροφή στην διεργασία πατέρα όπως και στο 1ο ερώτημα χρησιμοποίησα την συνάρτηση wait()

Ερώτημα 3ο //

Στο 3ο ερώτημα μας ζητήθηκε να αναπτύξουμε ένα shell script σε C όπου ο χρήστης θα δίνει σαν είσοδο ένα πρόγραμμα με πολλαπλές παραμέτρους, με την διαφορά

ότι τώρα θα υποστηρίζεται η χρήση pipes. Ο σχεδιασμός του script εμπεριέχει την υλοποίηση αρχικά ενός command line όπου έχουμε το dollar sign σαν αρχή του τερματικού και έναν έλεγχο με μία επανάληψη για το πότε τερματίζετε το script.Τα pipes είναι δίαυλοι όπου επιτρέπουν την επικοινωνία σε δύο η περισσότερες συγγενής διεργασίες και την ανταλλαγή δεδομένων. Επεκτείνοντας τα παραπάνω script δημιούργησα έναν ακόμα spliter για τον έλεγχο του input και την σύγκριση με το σύμβολο «|».Έπειτα για την υλοποίηση του προγράμματος δημιούργησα 2 child process ένα για την υλοποίηση του προγράμματος πριν τον δίαυλο και έναν για μετά. Προτίμησα την execvp για την εκτέλεση των διεργασιών και την dup2() για την επικοινωνία των διεργασιών και την δημιουργία αντιγράφου το δοθέντος descriptor.

Ερώτημα 4ο //

Στο 4ο ερώτημα αντιμετώπισα κάποια προβλήματα στην δημιουργία πολλαπλών pipes.Αρχικά δεν δούλευε το script για κανένα όρισμα παρά μόνο για το πρώτο, δηλαδή μέχρι ο χρήστη να πατήσει το « ».Προσπάθησα να αποθηκεύω κάθε φορά την δημιουργία της διεργασίας σε έναν μονοδιάστατο πίνακα και να το εκτελώ κομμάτι-κομμάτι αλλά δεν τα κατάφερα.