

1)Υπάρχει ένα αρχείο list.c μαζί με ένα list.h που περιέχουν κάποιες συναρτήσεις για λίστες.

2)Ως τρίτο αλγόριθμο για την εύρεση πρώτων αριθμών για να δω αν ο n είναι πρώτος κάνω το εξής: Βρίσκω χρησιμοποιώντας το κόσκινο του Ερατοσθένη όλους τους πρώτους μέχρι την τετραγωνική ρίζα του n . Στην συνέχεια ελέγχω αν κάποιος από αυτούς διαιρεί ακριβώς το n . Αν όχι τότε δεν είναι πρώτος.

3)Δεν έχει νόημα να δίνουμε ως όρισμα μεγάλο πλήθος παιδιών γιατί μετά από ένα σημείο απλά δημιουργούνται περιττές διεργασίες που απλά ελέγχουν συνεχώς ένα μηδενικό διάστημα για πρώτους αριθμούς. Επομένως αν ο ανώτατος ακέραιος που θα ελέγξουμε αν είναι πρώτος είναι μικρότερος από το τετράγωνο των παιδιών που έχουμε εισάγει ο αριθμός των παιδιών αλλάζει σε τετραγωνική ρίζα του ανώτατου ακεραίου.

4Η `roll` επιστρέφει -1 όταν κατά την εκτέλεσή της η διεργασία που την καλεί δέχεται κάποιο σήμα. Επομένως, όπως αναφέρθηκε στο `riaZZa`, το αγνοώ σε αυτήν την περίπτωση και διαβάζω κανονικά από το `pipe`.

5) Έχω δημιουργήσει ένα struct με πεδία prime και time. Στο prime αποθηκεύω τον πρώτο αριθμό που βρήκα και στο time τον χρόνο που μου πήρε να τον βρω. Για να βρω πόσο χρόνο είχε διάρκεια το κάθε φύλλο χρησιμοποιώ το παραπάνω struct. Σε αυτήν την περίπτωση στο time βάζω τον συνολικό χρόνο του κάθε worker και στο prime το του κάθε worker. Αυτά τα γράφω στο prime και τα διαβάζω από την ρίζα αφού έχουν τελειώσει όλα τα παιδιά και τα αποθηκεύω σε μια λίστα. Έχοντας ως δεδομένο ότι ένα φύλλο που εκτελείται πρώτα έχει μικρότερο pid από ένα που εκτελείται μετά ταξινομώντας την τελική λίστα με βάση το prime μπορώ να εκτυπώσω τους χρόνους με την σειρά.

6) Πολλές φορές το ένα παιδί παίρνει το διάστημα $[a,b]$ και το επόμενο το $[b,c]$. Επομένως το b θα ελεγχτεί 2 φορές και αν είναι πρώτος θα μπει 2 φορές στην λίστα. Επομένως, για αυτήν την περίπτωση υπάρχει μια συνάρτηση removeDuplicates που αφαιρεί όλα τα διπλότυπα από μια ταξινομημένη λίστα.