
ΛΕΙΤΟΥΡΓΙΚΑ ΣΥΣΤΗΜΑΤΑ II

ΕΑΡ. ΕΞΑΜΗΝΟ 2020-2021

Ασκήσεις Εργαστηρίου #1

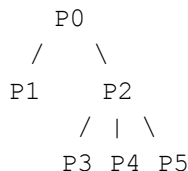
(Ημ. Παράδοσης: 18/4/2021)

1. Έστω ένα πρόγραμμα σε C στην αρχή του οποίου δίνονται οι εξής εντολές:

```
pid1=fork();  
if (pid1!=0) { pid2=fork(); pid3=fork(); }  
else { pid4=fork(); if (pid4==0) { pid5=fork(); } }  
.....
```

Πόσες διεργασίες παράχθηκαν με τις προηγούμενες εντολές; Ποια είναι η «συγγένεια» μεταξύ τους; Εξηγήστε συνοπτικά πως παρήχθησαν οι ανωτέρω διεργασίες (εντολή προς εντολή, με ποιο fork δημιουργήθηκε η κάθε μία κλπ). Δώστε επίσης ένα σχήμα στο οποίο θα φαίνεται με σαφή τρόπο η ιεραρχία μεταξύ των ανωτέρω διεργασιών.

2. Γράψτε ένα πρόγραμμα C με τις κατάλληλες εντολές **fork()** ώστε να δημιουργηθούν συνολικά (μαζί με το κυρίως πρόγραμμα) έξι διεργασίες, με την ακόλουθη δομή (δέντρο) συγγένειας:



Ως κύριο μέρος της εργασίας της κάθε διεργασία P_i θα πρέπει να τυπώνει απλά ένα μήνυμα στην οθόνη που να δηλώνει το όνομά της (P_i), το PID της και το PPID της. Το πρόγραμμά σας θα πρέπει να συμπεριλαμβάνει επιπλέον τις κατάλληλες εντολές αναμονής-επικοινωνίας ώστε να ικανοποιούνται οι ακόλουθοι περιορισμοί-απαιτήσεις: (α) η διεργασία P_0 πριν εκτελέσει το κύριο μέρος της εργασίας της να περιμένει την ολοκλήρωση της διεργασίας P_2 , και (β) η διεργασία P_2 πριν εκτελέσει την κύρια εργασία της να περιμένει την ολοκλήρωση όλων των παιδιών της. Επίσης η διεργασία P_0 πριν δημιουργήσει την P_2 να περιμένει να λάβει ένα μήνυμα 'hello from your first child' από τη διεργασία P_1 (και να το τυπώνει στην οθόνη). Τέλος, η διεργασία P_0 με το πέρας της εκτέλεσής της θα πρέπει να αντικαθίσταται από την εντολή ps.

3. Γράψτε ένα πρόγραμμα σε C το οποίο να υπολογίζει «παράλληλα» το άθροισμα των τετραγώνων των στοιχείων ενός διανύσματος A ($a_1^2+a_2^2+\dots+a_n^2$) [με τη βοήθεια 'p' threads όπου το κάθε thread θα υπολογίζει το επιμέρους άθροισμα 'n/p' όρων (τετραγώνων) του αθροίσματος – τα p, n, a, θα πρέπει να τα δίνει ο χρήστης ή να τα διαβάξετε από αρχείο – επίσης θεωρείστε ότι το 'n' είναι ακέραιο πολλαπλάσιο του 'p']. Πιο συγκεκριμένα, το κάθε thread θα πρέπει (α) να υπολογίζει τοπικά (σε μια τοπική μεταβλητή – π.χ. local_sum) το επιμέρους άθροισμα που του αναλογεί και (β) να ενημερώνει στο τέλος με το τοπικό του άθροισμα μια κοινή μεταβλητή (π.χ. total_sum) η οποία θα αντιπροσωπεύει το τελικό άθροισμα. Το τελικό αυτό άθροισμα θα πρέπει στο τέλος να τυπώνεται στην οθόνη. Υπάρχει κάπου κρίσιμο τμήμα/κώδικας που απαιτεί προστασία? Αν ναι, γράψτε επιπλέον τον κώδικα για να προστατευθεί το κρίσιμο αυτό τμήμα με χρήση του μηχανισμού 'κλειδώματος' που παρέχουν τα POSIX Threads.

Δοκιμάστε επίσης να τρέξετε το πρόγραμμά σας για μεγάλα και πολύ μεγάλα 'n' (χρησιμοποιώντας για input αντιστοίχου μεγέθους διανύσματα αρχικοποιημένα με τυχαίους αριθμούς) και μετρήστε τους χρόνους εκτέλεσης για ένα και περισσότερα από ένα threads (π.χ. 1, 2, 4 και 8 threads).

Παραδοτέα:

Κώδικας, σχολιασμός/τεκμηρίωση, ενδεικτικά τρεξίματα και χρόνοι εκτέλεσης.

Σχετικά με τη δομή και το format του/ων παραδοτέου/ων σας καλείστε να ακολουθήσετε τις οδηγίες που θα σας δώσει πιο συγκεκριμένα ο Καθηγητής του τμήματος σας.