
ΛΕΙΤΟΥΡΓΙΚΑ ΣΥΣΤΗΜΑΤΑ II

ΕΑΡ. ΕΞΑΜΗΝΟ 2024-2025

1^η Άσκηση Εργαστηρίου (12,5%)

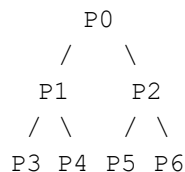
(Ημ. Παράδοσης: 27/4/2025)

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 (βήμα/εντολή προς βήμα/εντολή, με ποιο fork δημιουργήθηκε η κάθε μία / ποιο pid αντιστοιχεί σε 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) επτά διεργασίες, με την ακόλουθη δομή (δέντρο) συγγένειας:



Στη συνέχεια (αφού δημιουργηθούν δηλαδή όλες οι διεργασίες που υπαγορεύονται από το παραπάνω δέντρο), ως κύριο μέρος της εργασίας της κάθε διεργασία P_i θα πρέπει να τυπώνει ένα μήνυμα στην οθόνη που να δηλώνει το όνομά της (P_i), το PID της και το PPID της. Επιπλέον, το πρόγραμμά σας θα πρέπει να συμπεριλαμβάνει τις κατάλληλες εντολές αναμονής-επικοινωνίας ώστε να ικανοποιούνται οι ακόλουθοι περιορισμοί-απαιτήσεις:

(α) Η διεργασία P_0 πριν ολοκληρώσει την εκτέλεσή της θα πρέπει να περιμένει την ολοκλήρωση της διεργασίας P_2 . Με το πέρας της εκτέλεσής της η P_0 θα πρέπει επίσης να αντικαθίσταται από την εντολή *ps*.

(β) Η διεργασία P_1 πριν ολοκληρώσει την εκτέλεσή της θα πρέπει αρχικά να περιμένει να λάβει ένα μήνυμα 'hello from your child' από τη διεργασία P_3 (και να το τυπώνει στην οθόνη). Στη συνέχεια θα πρέπει να περιμένει επίσης την ολοκλήρωση της διεργασίας P_4 και να τυπώνει το exit value αυτής.

(γ) Η διεργασία P_2 πριν ολοκληρώσει την εκτέλεσή της θα πρέπει να περιμένει την ολοκλήρωση ενός εκ των δύο παιδιών της και να τυπώνει ποιο είναι αυτό (το PID δηλαδή της διεργασίας-παιδιού που ολοκληρώθηκε).

Επεκτείνετε επίσης το πρόγραμμά σας έτσι ώστε η διεργασία P_2 να δημιουργεί N διεργασίες/άμεσα παιδιά (αντί για τις δύο μόνο διεργασίες/άμεσα παιδιά P_5, P_6 που φαίνεται να δημιουργεί στο παραπάνω σχήμα), όπου το N θα δίνεται από το χρήστη. Κάθε μία από τις N διεργασίες που θα δημιουργηθούν θα πρέπει να τυπώνει απλά ένα διαγνωστικό μήνυμα με το PID της και το PPID της και να τερματίζει.

Είναι πιθανό κατά την εκτέλεση του προγράμματός σας να δημιουργηθούν orphan ή zombie διεργασίες; Και ποιες μπορεί να είναι αυτές; Σχολιάστε / τεκμηριώστε επαρκώς την απάντησή σας.

ΠΑΡΑΔΟΤΕΑ:

Τις απαντήσεις σας θα πρέπει να τις δώσετε σύμφωνα με τα υπαγορευόμενα στο συνοδευτικό αρχείο 'guidelines-ls2-ask1.pdf' (είναι αναρτημένο στο Eclass στον ίδιο κατάλογο με το αρχείο της παρούσας εκφώνησης).