Λειτουργικά Συστήματα 2021
2η Εργαστηριακή Άσκηση

Όνομα: Δημήτρης Βάσιος
Α.Μ. : el19404

Όνομα: Παύλος Καζάκος
Α.Μ. : el18403

Άσκηση 1

(1) Εάν τερματίσουμε πρόωρα την διεργασία Α η αρχική διεργασία που δημιούργησε το δέντρο τερματίζει και εμφανίζει το μήνυμα τερματισμού. Κάτι τέτοιο είναι λογικό καθώς η αρχική διεργασία main περιμένει τον τερματισμό της διεργασίας Α και όταν αυτό συμβεί εκτυπώνεται το κατάλληλο μήνυμα. Οι υπόλοιπες διεργασίες συνεχίζουν κανονικά την εκτέλεση. Όμως οι διεργασίες Β και C θα γίνου διεργασίες ζόμπι καθώς η διεργασία Α δεν θα τις περιμένει πλέον και άρα θα μεταβούν σε αυτή τη κατάσταση για ένα χρονικό διάστημα στο χώρο του πυρήνα του λειτουργικού συστήματος.

(2) Η συνάρτηση fork() επιστρέφει την τιμή της διεργασίας του παιδιού της διεργασίας που την καλεί ενώ η συνάρτηση getpid() επιστρέφει την τιμή της καλούσας διεργασίας. Επομένως σε αυτή τη περίπτωση το δέντρο διεργασιών είναι διαφορετικό.
Στην περίπτωση που καλέσουμε την συνάρτηση show_pstree(getpid()) τότε θα εμφανιστούν δύο ακόμα διεργασίες, η sh και η pstree καθώς και η αρχική διεργασία main.
Αυτό συμβαίνει καθώς στην main καλούμε την συνάρτηση show_pstree(), η οποία δημιουργεί μία διεργασία παιδί στην οποία καλούμε ένα shell command.

(3) Καθε επεξεργαστής μπορεί να εκτελεί πεπερασμένο αριθμό διεργασιών ταυτόχρονα, Καθε Λειτουργικό Σύστημα αναλογα σε ποιο υπολογιστικό συστημα απευθυνεται αρα και σε επεξεργαστική ισχύ θέτει τα αναλογο οριο οσον αφορα τον μέγιστο αριθμό ταυτόχρονων διεργασιων που μπορούν να εκτελούνται. Ετσι το συστημα απαλλάσσεται από τον κίνδυνο απο το να ερθει σε κορεσμό η να εκτελέσει καποιο πρόγραμμα λανθασμένα ή με μη λειτουργικό τρόπο.

Άσκηση 2

(1) Τα μηνύματα έναρξης εμφανίζονται κατά βάθος έως ότου φτάσουν σε κόμβο που δεν έχει παιδιά. Τότε η διεργασία κάνει sleep και στη συνέχεια τερματίζει και φαίνεται το αντίστοιχο μήνυμα τερματισμού. Στη συνέχεια πάλι εφαρμόζοντας τον αλγόριθμο κατά βάθος ελέγχουμε το αμέσως επόμενο κόμβο, τυπώνουμε μήνυμα έναρξης, αν δεν έχει παιδιά κάνει sleep και εμφανίζεται μήνυμα τερματισμού. Όταν τερματίζουν όλοι τα παιδιά ενός κόμβου τότε τερματίζεται και ο ίδιος ο κόμβος και τυπώνεται το αντίστοιχο μήνυμα τερματισμού.

Άσκηση 3

(1) Με τη χρήση των σημάτων αντί της συνάρτησης sleep() έχουμε τον πλήρη έλεγχο των διεργασιών. Η χρήση της συνάρτησης sleep είναι βοηθητική αλλά δυσκολεύει τον συγχρονισμό μεταξύ των διεργασιών καθώς πρέπει να γνωρίζουμε επακριβώς του χρόνους που είναι ανοιχτή μία διεργασία. Με τη χρήση των σημάτων όμως μπορούμε να προσδιορίσουμε επακριβώς πότε θέλουμε μία διεργασία να σταματήσει ή να συνεχίσει από εκεί που σταμάτησε.

(2) Με τη χρήση της συνάρτησης wait_for_ready_children() εξασφαλίζουμε ότι η διεργασία πατέρα θα περιμένει έως ότου το παιδί της θα αναστάλλει τη λειτουργία του και δεν θα προχωρήσει στο επόμενο παιδί. Επιπλέον μας ενημερώνει και μας εκτυπώνει την κατάσταση στην οποία βρίσκεται το παιδί.

Άσκηση 4

(1) Είναι εφικτό μία διεργασία πατέρας να χρησιμοποιεί μόνο μία σωλήνωση για όλες τις διεργασίες παιδιά. Τη πρώτη φορά γράφουμε στη σωλήνωση στην αρχή του file descriptor και την δεύτερη φορά για το επόμενο παιδί γράφουμε στο ίδιο file descriptor αλλά αμέσως μετά από κει που έγραψε το πρώτο. Με αυτόν τον τρόπο μπορούμε να πετύχουμε το ίδιο αποτέλεσμα με μία μόνο σωλήνωση.

(2) Σε ένα σύστημα παράλληλων επεξεργαστών οι διεργασίες καταμερίζονται αντίστοιχα στους επεξεργαστές και παράγουμε αποτέλεσμα πιο γρήγορα από το να εκτελείται η διεργασία σειριακά. Στο δέντρο διεργασιών κάνουμε ακριβώς αυτό. “Σπάμε” το πρόγραμμα σε επιμέρους διεργασίες ώστε να καταμερίσουμε το βάρος του συνολικού προγράμματος και έχουμε αποτέλεσμα πιο γρήγορα και πιο αποδοτικά από ότι άμα επιλέγαμε να το κάνουμε με μία μόνο διεργασία.