

Ενδιάμεσος Διαγωνισμός

Με ένα φύλλο Α4. Απαντήστε όλα τα θέματα. Διάρκεια εξέτασης: Εβδομήντα λεπτά.

1. (15 + 15 = 30 μονάδες) Το ζήτημα αυτό αναφέρεται στα κατανεμημένα προγράμματα C *display.c*, *prog1.c* και *prog2.c* που φαίνονται παρακάτω:

```
/* display.c */
main ( )
{
    int i;
    for (i = 0; i < 2; i++)
        printf("Distributed System\n");
}

/* prog1.c */
main ( )
{
    int i,childpid,status;
    for (i = 0; i < 5; i++)
    {
        childpid = fork( );
        if (childpid < 0)
        {
            printf("Error with fork.\n");
            exit(1);
        }
        else
            if (childpid == 0)
            {
                if (execlp("display", "display", (char *)0) == -1)
                    printf("Error with exec.\n");
                exit(2);
            }
    }
}

/* prog2.c */
main ( )
{
    int i,childpid,status;
    for (i = 0; i < 5; i++)
    {
        childpid = fork( );
        if (childpid < 0)
        {
            printf("Error with fork.\n");
            exit(1);
        }
        else
            if (childpid > 0)
            {
                if (execlp("display", "display", (char *)0) == -1)
                    printf("Error with exec.\n");
                exit(2);
            }
    }
}
```

- (α) Πόσες διεργασίες θα δημιουργηθούν όταν εκτελεστεί ο κώδικας του *prog1.c*; Πως θα σχετίζονται μεταξύ τους οι διεργασίες αυτές; Για να περιγράψετε τη σχέση τους, μπορείτε να χρησιμοποιήσετε ένα διάγραμμα.
- (β) Πόσες διεργασίες θα δημιουργηθούν όταν εκτελεστεί ο κώδικας του *prog2.c*; Πως θα σχετίζονται μεταξύ τους οι διεργασίες αυτές; Για να περιγράψετε τη σχέση τους, μπορείτε να χρησιμοποιήσετε ένα διάγραμμα.

Εξηγείστε πλήρως τις απαντήσεις σας. Βεβαιωθείτε ότι έχετε καλύψει τις περιπτώσεις που όλες ή μερικές κλήσεις του συστήματος επιτύχουν ή αποτύχουν.

2. (20 μονάδες) Παρουσιάστε ένα καταναμημένο πρόγραμμα C με τα εξής χαρακτηριστικά:

- Αμέσως μετά την εκκίνησή του, το πρόγραμμα δημιουργεί 2 παιδιά.
- Η πρώτη διεργασία-παιδί θα δημιουργήσει 2 παιδιά, ενώ η δεύτερη διεργασία-παιδί θα δημιουργήσει 6 παιδιά.
- Καθεμιά από τις νέες διεργασίες-παιδιά θα υπολογίζει μια σταθερά y έχοντας διαφορετικό χρόνο στη διάθεσή της, ως εξής: Οι δύο διεργασίες που δημιουργούνται από την πρώτη διεργασία-παιδί έχουν στη διάθεσή τους 1 sec και 2 sec, αντίστοιχα. Οι έξι διεργασίες που δημιουργούνται από τη δεύτερη διεργασία-παιδί έχουν στη διάθεσή τους 1 sec, 2 sec, 3 sec κ.ο.κ.
- Μετά το πέρας του χρόνου της κάθε διεργασίας-παιδιού, θα τυπώνεται η τιμή του y που υπολόγισε η διεργασία έχοντας στη διάθεσή της τό χρόνο αυτό.
- Η διεργασία-πατέρας θα περιμένει όλες τις διεργασίες-παιδιά να ολοκληρωθούν, και μετά θα τερματίσει.

Ο υπολογισμός της σταθεράς y θα βρίσκεται σε ένα ξεχωριστό πρόγραμμα (π.χ., *compute-y*), το οποίο θα δέχεται ως όρισμα το διαθέσιμο χρόνο υπολογισμού. Το πρόγραμμα αυτό θα καλείται μέσω της κλήσης συστήματος *execlp*. Θεωρείστε ότι η συνάρτηση για τον υπολογισμό της σταθεράς y είναι ήδη ορισμένη με το όνομα *calculate - y*.

3. ($5 \times 10 = 50$ μονάδες) Θεωρείστε ένα καταναμημένο σύστημα το οποίο αποτελείται από δύο επεξεργαστές και δύο μηχανές. Κάθε ένας από τους επεξεργαστές φέρει μοναδιαίο φορτίο, ενώ οι δύο μηχανές έχουν ταχύτητες c_1 και c_2 , αντίστοιχα.

- (α) Προσδιορίστε σύστημα εξισώσεων (και ανισοτήτων) για την Πλήρως Μικτή Ισορροπία Nash του συστήματος.
- (β) Επιλύστε το σύστημα ώστε να προσδιορίσετε τις σχετικές πιθανότητες για την Πλήρως Μικτή Ισορροπία Nash. Για ποιές τιμές των c_1 και c_2 υπάρχει η Πλήρως Μικτή Ισορροπία Nash;
- (γ) Παρουσιάστε αλγεβρική έκφραση για το Κοινωνικό Κόστος της Πλήρως Μικτής Ισορροπίας Nash που προσδιορίσατε.
- (δ) Βρείτε κατάλληλες τιμές των c_1 και c_2 που μεγιστοποιούν (αντίστοιχα, ελαχιστοποιούν το Κοινωνικό Κόστος που προσδιορίσατε.
- (ε) Προσδιορίστε τα καλύτερα κάτω και άνω φράγματα που μπορείτε για το Κόστος Αναρχίας του συστήματος.