

## ΛΕΙΤΟΥΡΓΙΚΑ ΣΥΣΤΗΜΑΤΑ ΕΡΓΑΣΙΑ 3

### 1. Εντολές μεταγλώττισης και εκτέλεσης:

make all δημιουργεί το εκτελέσιμο

./PS.o param1 param2 param3 param4 param5 param6 param7

Τα param1-6 είναι οι παράμετροι που ζητούνται από την εκφώνηση με τη σειρά που ζητούνται:

param1 : μέσος χρόνος μεταξύ διαδοχικών αφίξεων διεργασιών (εκθετικά κατανεμημένος χρόνος)

param2: μέση διάρκεια ζωής διεργασίας (εκθετικά κατανεμημένη διάρκεια)

param3: μέση διάρκεια εκτέλεσης κρίσιμης περιοχής (εκθετικά κατανεμημένη διάρκεια)

param4: μέγιστο πλήθος διεργασιών προσομοιωτή

param5: πιθανότητα εισόδου σε κρίσιμη περιοχή ανά χρονοθυρίδα (k)

param6: πλήθος σηματοφόρων (S).

Η param7 είναι extra και θα μας δίνει το value της χρονοθυρίδας

make clear διαγράφει το εκτελέσιμο

### 2. Τεκμηρίωση

Το πρόγραμμα αποτελείται από 4 modules:

Semaphore ,PCB(process control block), Utils, Process Scheduler

Η κλάση του σηματοφόρου προσομοιώνει την λειτουργία του σηματοφόρου συστήματος με συναρτήσεις down/up και push και pop στο queue

Το utils περιλαμβάνει βοηθητικές συναρτήσεις για την χρήση που χρειαζόμαστε (επιλογή τυχαίου αριθμού εκθετικής κατανομής και επιλογή τυχαίου αποτελέσματος σύμφωνα με δοσμένη πιθανότητα)

Η κλάση PCB προσομοιώνει τη λειτουργία μιας διεργασίας και περιλαμβάνει τις μεταβλητές

```

class PCB
{
private:
    int pid;                // process PID
    int priority;           // process priority
    bool access;            // has access to critical section
    state process_state;    // Process state
    double burst_time;      // total time to finish
    double remaining_time;  // remaining time to finish CS
    double arrival_time;    // time of arrival
    double critical_section_time; // time that process requires in critical section
    int locked_sem;         // Requested Semaphore (or locked)
}

```

οι οποίες θα μας βοηθήσουν να προσομοιώσουμε τον scheduler

Τέλος η κλάση process scheduler είναι αυτή που προσομοιώνει την χρονοδρομολόγηση. Οι functions που χρησιμοποιεί είναι οι:

- Check incoming()

Ελέγχει σε κάθε χρονοθυρίδα αν υπάρχουν νέες αφίξεις διεργασιών ώστε να τις προσθέσει την ουρά αναμονής προτεραιότητας.

- Run Process()

Δεδομένου ενός process αποφασίζει με ποιο τρόπο θα τρέξει αυτό, ανάλογα αν χρειάζεται να μπει σε critical section , αν είναι ήδη μέσα ή απλή εκτέλεση.

- Run InCritical()

Η συγκεκριμένη συνάρτηση χειρίζεται την εκτέλεση της διεργασίας σε περίπτωση που θα πρέπει να εκτελέσει στο critical section. (εδώ κάνουμε και handle λάθη που μπορεί να προκύψουν)

- Print\_stats()

Εκτυπώνει στατιστικά για τις διεργασίες

Συνολικά η κεντρική ιδέα λειτουργίας του προγράμματος σε μια χρονοθυρίδα:

Ο scheduler κοιτάει για αφίξεις και προσθέτει τις διεργασίες στο priority queue.

Ο scheduler τρέχει τη διεργασία στο front του PQ με τη μεγαλύτερη προτεραιότητα και την βγάζει από το PQ. Σε περίπτωση που μια διεργασία καταφέρει να ζητήσει access στον σηματοφόρο υπάρχουν δύο επιλογές. Είτε θα καταφέρει να τον κάνει down είτε θα μπει στο

queue του σημαφόρου και ανταυτού θα τρέξει η διεργασία που είναι πρώτη στην ουρά. Τέλος μόλις τερματίσει την λειτουργία στο CS θα απελευθερώσει τον σημαφόρο. Τελικά όταν μια διεργασία τερματίσει τίθεται σε status terminated και δεν ξαναμπαίνει στο PQ.

Σε περίπτωση που για κάποιο λόγο μια διεργασία τερματίσει και δεν έχει προλάβει να κάνει up τον αντίστοιχο semaphore, θα το κάνει ο scheduler ώστε να μην έχουμε deadlock.

Πρόβλημα:

Λόγο κάποιου bug που δεν έχω καταφέρει να βρω τα waiting times σε semaphores έρχονται μηδενικά δυστυχώς.