
Άσκηση 2

Semaphores: **WorkersQueue**->Ουρά αναμονής διαθέσιμων εργατών.

WaitingMain->Αναμονή της main μέχρι να υπάρχει τουλάχιστον ένας worker διαθέσιμος

TerminatingMain->Αναμονή της main για τερματισμό όλων των workers.

mtx->Αμοιβαίος αποκλεισμός threads και main για τα κρίσιμα σημεία.

Global: **WaitingWorkers, Terminating, TerminatedWorkers, flag**(=totalWorkers), **times**

Main:

(Κάθε worker είναι μια θέση ενός πίνακα από structs.)

For(){δημιουργία Worker και ανάθεση σε αυτόν του πρώτου αριθμού}

While(1){

 Scanf(number); if(EOF){break;}

 Down(WaitingMain);

 For(){ εύρεση διαθέσιμου εργάτη με έλεγχο της μεταβλητής του struct isready==1

 Ανάθεση τιμής σε αυτόν και αλλαγή του isready=0}

}

 down(mtx); Terminating=1;

/*το ελεγχουμε δυο φορές σε περίπτωση που οι τελευταίοι workers δεν προλάβαν να ξεκινήσουν*/

If(τουλάχιστον ένας worker δουλεύει ακόμη){ up(mtx); Down(WaitingMain);}

Terminating=-2;

For(totalWorkers){up(WorkersQueue); down(WaitingMain);}

Down(TerminatingMain);

Workers:

While(1){

 WaitingWorkers++; Isready=1; times++;

 If(Terminating==1 & τελευταίος worker που ολοκλήρωσε δουλειά){up(WaitingMain);}

 up(mtx);

 While(1){

```
    Down(WorkersQueue);

    If(Terminating==2){   terminatedWorkers++;

        If(terminatedWorkers==TotalWorkers){up(TerminatingMain);}

        up(WaitingMain);

        Return();}

    }

    If(ανάθεση νέου αριθμού από main στον συγκεκριμένο worker){break;}

    Else{up(WorkersQueue);}

}

down(mtx);   WaitingWorkers--;

If(WaitingWorkers>0){up(WaitingMain);}   up(mtx);


down(mtx);   Primetest();

}
```