

# Implementation of “Shortest Remaining Time” scheduling policy in the Linux Operating System

Σε αυτή την άσκηση σας ζητείτε να υλοποιήσετε τον αλγόριθμο “*Shortest Remaining Time*” στον scheduler του λειτουργικού συστήματος Linux. Θεωρήστε ότι ο χρόνος είναι χωρισμένος σε κβάντα. Στην αρχή κάθε κβάντου ο scheduler σκοτώνει όλες τις διεργασίες που έχουν ξεπεράσει το hard deadline τους. Για κάθε διεργασία που έχει περάσει το soft deadline αλλά όχι το hard deadline ο scheduler διαλέγει πρώτη αυτή που έχει το μικρότερο εναπομείναντα χρόνο (*shortest remaining time*). Για τις υπόλοιπες διεργασίες, δηλαδή αυτές που δεν έχουν ξεπεράσει το soft deadline τους, επιλέγεται εκείνη με την μεγαλύτερη προτεραιότητα (*highest priority first*). Τέλος, θα πρέπει να δοκιμάσετε τις αλλαγές σας στον scheduler με ένα demo πρόγραμμα.

## Τροποίσεις στον Linux Kernel

Σε αυτή την άσκηση θα χρησιμοποιήσετε τον qemu emulator και το disk image της προηγούμενης άσκησης για να ξεκινήσετε το λειτουργικό σύστημα. Τις οδηγίες για το πώς θα κάνετε compile τον Linux kernel και πως μπορείτε να κάνετε boot το image, μπορείτε να τις ξαναδείτε στην περιγραφή της προηγούμενης άσκησης. Στην υλοποίηση της άσκησης αυτής θα ξεκινήσετε χρησιμοποιώντας τον κώδικα σας από την άσκηση 3.

Ο κώδικας του Linux scheduler είναι στο αρχείο `linux-source-2.6.38.1/kernel/sched.c`. Η κυριότερη συνάρτηση του Linux scheduler είναι η :

```
asmlinkage void __sched schedule(void).
```

Τα βήματα που πρέπει να ακολουθήσετε τροποποιώντας τον kernel περιλαμβάνουν τα εξής:

1. Ο context switching μηχανισμός εκτελείτε σε συγκεκριμένα κβάντα χρόνου.
2. Ο μηχανισμός θα ενεργοποιείται στην αρχή κάθε κβάντου.
3. Σε κάθε κβάντο, ο scheduler θα αναζητά όλες τις διεργασίες που έχει περάσει το hard deadline τους και θα τις σκοτώνει.
4. Ο context switch μηχανισμός θα ελέγχει σε κάθε κβάντο εαν υπάρχουν ενεργές διεργασίες που έχει περάσει το soft deadline τους αλλά όχι το hard deadline τους. Σε αυτή τη περίπτωση θα επιλέγεται η διεργασία με το πιο μικρότερο εναπομιναντα χρόνο.
5. Για όλες τις υπόλοιπες διεργασίες που δεν έχουν περάσει το soft deadline τους επιλέγουμε αυτή με την μεγαλύτερη προτεραιότητα.

## Τι πρέπει να παραδώσετε

1. Το καινούριο kernel image, δηλαδή το αρχείο linux-2.6.38.1/arch/x86/boot/bzImage.
2. Όλα τα αρχεία που τροποποιήσατε ή δημιουργήσατε στον source code του Linux kernel 2.6.38.1 για να υλοποιήσετε το scheduling policy. Δηλαδή όλα τα αρχεία .c, .h, Makefile κτλ που κάνατε κάποια αλλαγή ή δημιουργήσατε εσείς. Μην παραδώσετε αρχεία που δεν χρειάστηκε να τα τροποποιήσετε για την υλοποίησή σας
3. Το source code από όλα τα test προγράμματα που γράψατε και τρέξατε μέσα στο guest Linux OS για να δοκιμάσετε το scheduling policy που υλοποιήσατε. Και επιπλέον ότι header files χρησιμοποιήσατε για type και function definitions ( πχ το unist.h). Δηλαδή τα αρχεία .c, .h και Makefile καθώς και ότι άλλο αρχείο δημιουργήσατε στο guest OS για να δοκιμάσετε τις αλλαγές σας.
4. Ένα README στο οποίο να περιγράφετε συνοπτικά (αλλά περιεκτικά και ξεκάθαρα) όλα τα βήματα που ακολουθήσατε για την δημιουργία του scheduling policy. Επίσης πρέπει να σχολιάσετε τι παρατηρήσατε απο τα test προγράμματα που τρέξατε. Αν έχετε κάνει κάτι διαφορετικό ή παραπάνω απο όσα αναφέρουμε στην εκφώνηση της άσκησης σε οποιοδήποτε βήμα μπορείτε επίσης να το αναφέρετε στο README. Καλό θα ήταν το README να είναι από 20 μέχρι 30 γραμμές.