

Implementation of "Shortest Period First" scheduling policy in the Linux Operating System

Σε αυτή την άσκηση σας ζητείται να υλοποιήσετε τον αλγόριθμο "Shortest Period First" στον scheduler του λειτουργικού συστήματος Linux. Κάθε διεργασία δηλώνει στον χρονοπρογραμματιστή ποια είναι η περίοδος της αλλά και πόσο χρόνο εκτέλεσης (execution time) χρειάζεται μέσα στην περίοδο αυτή. Σε κάποια χρονική περίοδο, υπάρχει περίπτωση να πρέπει να εκτελεστούν πολλαπλές διεργασίες. Σε τέτοιες περιπτώσεις ο χρονοπρογραμματιστής δίνει προτεραιότητα αντιστρόφως ανάλογη με την δηλωμένη περίοδο κάθε διεργασίας. Μετά την εκτέλεση της, μια διεργασία δεν μπορεί να εκτελεστεί ξανά στην ίδια περίοδο. Τέλος, κάθε διεργασία ορίζει τον αριθμό περιόδων που θέλει να εκτελεστεί, όπως έχει περιγραφεί στην προηγούμενη άσκηση.

Τροποποιήσεις 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:

- Ο scheduler βρίσκει την διεργασία με τη μεγαλύτερη προτεραιότητα και την εκτελεί. Κάθε διεργασία εκτελείται ακριβώς μια φορά ανά περίοδο.
- Αφαιρεί από τον αριθμό των περιόδων που πρέπει να εκτελεστεί η διεργασία, όταν η διεργασία ολοκληρώσει τον χρόνο εκτέλεσής της ανά περίοδο.
- Ελέγχει αν μια περίοδος έχει τελειώσει για να προγραμματίσει ξανά τις διεργασίες που μένουν.
- Ελέγχει αν κάποια διεργασία έχει μεγαλύτερη προτεραιότητα από την διεργασία που εκτελείται. Σε αυτή την περίπτωση η διεργασία υψηλής προτεραιότητας πρέπει να διακόψει τη διεργασία χαμηλής προτεραιότητας, όταν η περίοδος της ξεκινήσει. Η διεργασία που διεκόπη θα πρέπει να συνεχίσει μετά.
- Αν ο αριθμός περιόδων μιας διεργασίας φτάσει στο 0 πρέπει να αφαιρεθεί από τη λίστα με τις προγραμματισμένες διεργασίες.

Παράδειγμα εκτέλεσης

Έστω δύο διεργασίες, Δ0 με περίοδο 10 δευτερόλεπτα και διάρκεια 5 δευτερόλεπτα και Δ1 με περίοδο 3 δευτερόλεπτα και διάρκεια 1 δευτερόλεπτο. Ο δρομολογητής τρέχει στις εξής στιγμές:

- 0: Δ1
- 1: Δ0
- 3: τρέχει την Δ1 και σταματά την Δ0 η οποία έχει τρέξει για 2 από τα 5 δευτερόλεπτά της
- 4: Δ0
- 6: τρέχει την Δ1 και σταματά την Δ0 η οποία τώρα έχει τρέξει για 4 από τα 5 δευτερόλεπτά της
- 7: Δ0
- 8: δεν υπάρχει καμία να τρέξει
- 9: Δ1

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 τοunistd.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 περιγραφητέ συνοπτικά (αλλά περιεκτικά και ξεκάθαρα) όλα τα βήματα που ακολουθήσατε για την δημιουργία του scheduling policy. Επίσης πρέπει να σχολιάσετε τι παρατηρήσατε από τα test προγράμματα που τρέξατε. Αν έχετε κάνει κάτι διαφορετικό ή παραπάνω από 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.