Εθνικό και Καποδιστριακό Πανεπιστήμιο Αθηνών
Τμήμα Πληροφορικής και Τηλεπικοινωνιών
Λειτουργικά Συστήματα (Κ22) / Περίοδος 2023-2024
2η Εργασία

Υποστήριξη priority scheduling xv6

Ο scheduler στο xv6 είναι απλός round robin που εκτελεί τις διεργασίες κυκλικά. Σε αυτή την εργασία, θα υλοποιηθεί ένας scheduler που υποστηρίζει προτεραιότητες, δηλαδή εκτελεί τις διεργασίες με υψηλότερη προτεραιότητα πρώτα και στη συνέχεια εκτελεί τις διεργασίες με χαμηλότερη προτεραιότητα.

Εισαγωγή νέων κλήσεων συστήματος

Για την εισαγωγή νέων κλήσεων συστήματος στο xv6, θα χρειαστεί να κατανοήσετε και να τροποποιήσετε τα παρακάτω αρχεία::

proc.c, proc.h, syscall.c, syscall.h, sysproc.c, user.h, and usys.pl.

Συγκεκριμένα:

- `user.h` - περιέχει τις δηλώσεις των κλήσεων συστήματος..
- `usys.pl` - παράγει μία λίστα με το σύνολο των κλήσεων συστήματος που εξάγει ο πυρήνας.
- `syscall.h` - περιέχει την αντιστοίχιση ονόματος κλήσης συστήματος σε αριθμό κλήσης συστήματος. Πρέπει να προστεθούν νέοι αριθμοί για νέες κλήσεις συστήματος
- `syscall.c` - περιέχει βοηθητικές συναρτήσεις για τα ορίσματα των κλήσεων συστήματος και δείκτες προς τις υλοποιήσεις τους
- `sysproc.c` - περιέχει υλοποιήσεις των κλήσεων συστήματος που σχετίζονται με διεργασίες.
- `proc.h` - περιέχει τη δομή `struct proc`. Θα χρειαστεί να επεκταθεί ώστε να αποθηκεύει επιπλέον πληροφορίες για κάθε διεργασίες που θα είναι απαραίτητες
- `proc.c` - περιέχει τη συνάρτηση `scheduler()` που υλοποιεί τον χρονοπρογραμματισμό και την αλλαγή μεταξύ διεργασιών

Θα υλοποιήσετε τις εξής νέες κλήσεις συστήματος στο xv6

1. Κλήση συστήματος `int setpriority(int num)`. Θέτει την προτεραιότητα για την καλούσα διεργασία ίση με `num`. Οι επιτρεπτές τιμές της προτεραιότητας είναι από 1 (υψηλότερη προτεραιότητα) ως 20 (χαμηλότερη προτεραιότητα). Η προκαθορισμένη προτεραιότητα είναι 10.
2. Κλήση συστήματος `int getpinfo(struct pstat *)`. Μέσω αυτής της κλήσης συστήματος θα αντιγράφει στο `struct pstat *` πληροφορίες για κάθε ενεργή διεργασία (pid, ppid, όνομα, προτεραιότητα, κατάσταση, μέγεθος και ό,τι άλλο μπορεί να χρειαστεί). Η δομή αυτή θα οριστεί σύμφωνα με την `struct proc`.

Για επιτυχή εκτέλεση επιστρέφουν οι κλήσεις συστήματος επιστρέφουν 0, αλλιώς επιστρέφουν -1.

Για την υποστήριξη προτεραιότητας, θα πρέπει να επεκταθεί η `struct proc` ώστε να υποστηρίζει προτεραιότητες.

Υλοποίηση προγράμματος χρήστη ps

Το πρόγραμμα σας θα υλοποιεί παρόμοια λειτουργία με την εντολή `ps` στο Linux, δηλαδή θα δείχνει πληροφορίες σχετικά με τις ενεργές διεργασίες στο xv6. Για το λόγο αυτό θα υλοποιήσετε το `ps.c` πρόγραμμα στον κατάλογο `user`, που να χρησιμοποιεί την κλήση συστήματος `getpinfo(struct pstat *)`. Στη συνέχεια θα χρησιμοποιείται από το πρόγραμμα η δομή `struct pstat` που έχει γεμίσει από την κλήση συστήματος.

Υλοποίηση priority based scheduler

Ο χρονοπρογραμματιστής (scheduler) στο xv6 υλοποιείται στη συνάρτηση `scheduler()` στο αρχείο proc.c. Η συνάρτηση αυτή καλείται από τη συνάρτηση `main()` στο `main.c` ως το τελευταίο βήμα της αρχικοποίησης. Η συνάρτηση `scheduler()` δεν επιστρέφει ποτέ. Επαναλαμβάνεται για πάντα χρονοπρογραμματίζοντας την επόμενη διαθέσιμη διεργασία προς εκτέλεση.

Η διαφορά μεταξύ του πρωτότυπου και του scheduler που καλείστε να υλοποιήσετε είναι ο τρόπος που θα επιλέγεται η επόμενη διεργασία. Ο νέος χρονοπρογραμματιστής θα ψάχνει σε όλες τις διεργασίες εκείνη που είναι `RUNNABLE` και έχει την υψηλότερη προτεραιότητα (αντί για την επόμενη που είναι σε status `RUNNABLE`). Αν υπάρχουν περισσότερες διεργασίες με την ίδια προτεραιότητα, τις χρονοπρογραμματίζει κυκλικά (round-robin). Ένας τρόπος να το κάνει αυτό είναι να αποθηκεύει την τελευταία διεργασία και να ξεκινά από εκεί για να ψάξει όλες τις διεργασίες.

Όταν ολοκληρώσετε την εργασία, ο πυρήνας θα πρέπει να περνάει όλα τα τεστ στα προγράμματα `usertests`, δηλαδή:

```
$ usertests
...
ALL TESTS PASSED
$
```

Κώδικας

Χρησιμοποιήσετε τον κώδικα για την εργασία ως εξής:

```
$ git clone git://gallagher.di.uoa.gr/xv6-project-2023
Cloning into 'xv6-project-2023'...
...
```

```
$ cd xv6-project-2023
```

Το αποθετήριο (repository) xv6-project-2023 περιέχει τον ίδιο κώδικα με το βασικό αποθετήριο του xv6 https://github.com/mit-pdos/xv6-riscv.

Τα αρχεία που θα χρειαστείτε για αυτή την εργασία διανέμονται μέσω του συστήματος ελέγχου πηγαίου κώδικα git. Μπορείτε να βρείτε πληροφορίες για το git στο βιβλίο git ή σε άλλες δημόσιες πηγές. Το git επιτρέπει να διατηρείτε πληροφορία για όλες τις αλλαγές που έχετε κάνει στον κώδικα. Για παράδειγμα, αν τελειώσετε ένα μέρος της εργασίας και θέλετε να καταχωρήσετε τοπικά τις αλλαγές σας, μπορείτε να καταγράψετε (commit) τις αλλαγές σας μέσω της εντολής

```
$ git commit -am 'my solution for k22 project'
Created commit 60d2135: my solution for k22 project
 1 files changed, 1 insertions(+), 0 deletions(-)
$
```

Ημερομηνία Παράδοσης: 21 Ιαν 2024
Τρόπος παράδοσης: υποβολή στο eclass, θα πρέπει να παραδοθεί ένα αρχείο tar με περιεχόμενο όλα τα σχετικά αρχεία.
Συνοδευτικό υλικό: τεκμηρίωση 3-4 σελίδων που να εξηγεί τον τρόπο με τον οποίο εργαστήκατε.
Υλοποίηση: η εργασία είναι ατομική.
Η εργασία θα εξεταστεί σε συμβατό xv6 emulator (Linux, Windows WSL) με πρόγραμμα που θα ανακοινωθεί μετά την ημερομηνία παράδοσης.