Κλήση συστήματος int setpriority(int num):

1) proc.h:
-Προσθήκη πεδίου int priority στο struct proc.

2) sysproc.c:

-sys_setpriority(void), κλήση setpriority(priority), το οποίο όρισμα μπορεί να ληφθεί, μέσω της συνάρτησης argint.

3) proc.c:

-allocproc(void), γίνεται αρχικοποίηση του priority με την default τιμή που είναι το 10.
-setpriority(int priority), αρχικοποίηση του priority της τρέχουσας διεργασίας με την τιμή priority και επιπλέον έλεγχος για τυχόν λάθη τιμών στο priority.

4) Επιπλέον προσθήκες στα αρχεία, οι οποίες χρειάζονται, για τη δημιουργία οποιασδήποτε κλήσης συστήματος.

Κλήση συστήματος int getpinfo(struct pstat*):

1) pstat.h:
-Header file στο οποίο υπάρχει η δομή του, struct pstat.

2) sys call.c:

-argpstat(int, struct pstat*, int), έλεγχος εάν είναι δυνατή η αρχικοποίηση του struct pstat. Εάν υπάρχει διαθέσιμος χώρος, τελικά γίνεται η αρχικοποίηση, διαφορετικά επιστρέφει -1.

3) sysproc.c:

-sys_getpinfo(void), αρχικοποίηση struct pstat με τη συνάρτηση argpstat, και κλήση συνάρτησης, getpinfo(&pstats).

4) proc.c:

-getpinfo(struct pstat*), διατρέχονται όλες οι διεργασίες οι οποίες δεν είναι unused και αποθηκεύονται στο struct pstat, τα κατάλληλα δεδομένα, που θα είναι χρήσιμα (PID, PPID, PRIORITY, NAME, STATE).
-Χρήση συνάρτησης, copyout, ώστε τα δεδομένα του struct pstat, να μεταφερθούν, από kernel level σε user level.

Υλοποίηση προγράμματος χρήστη ps

1) -Χρήση πιο πάνω κλήσης συστήματος getpinfo(), ώστε τα απαραίτητα, δεδομένα που χρειάζονται να τυπωθούν, να είναι στο struct pstat.

2) -Εκτύπωση, των πεδίων, του πιο πάνω struct.

Υλοποίηση priority-based scheduler

1) proc.c
-Στη συνάρτηση scheduler, εύρεση υψηλότερης προτεραιότητας, μεταξύ των διεργασιών που είναι runnable.

2) Αντιστοίχιση, τρέχουσας υψηλότερης προτεραιότητας, με τις διεργασίες που έχουν ίση προτεραιότητα (δηλαδή την υψηλότερη), όπου και τελικά εκτελούνται.

Tests (υπολογιστές linux της σχολης):

1) usertests:

```
OK
test sbrklast: OK
test sbrk8000: OK
test badarg: OK
usertests slow tests starting
test bigdir: OK
test manywrites: OK
test badwrite: OK
test execout: OK
test diskfull: balloc: out of blocks
ialloc: no inodes
ialloc: no inodes
OK
test outofinodes: ialloc: no inodes
OK
ALL TESTS PASSED
```

2) priotest (make CPUS=1 qemu):

```
Child pid 19, small, with priority 2 finished. Useless sum: 448743748
Child pid 38, small, with priority 2 finished. Useless sum: 448743748
Child pid 39, small, with priority 3 finished. Useless sum: 448743748
Child pid 20, small, with priority 3 finished. Useless sum: 448743748
Child pid 40, small, with priority 4 finished. Useless sum: 448743748
Child pid 21, small, with priority 4 finished. Useless sum: 448743748
Child pid 41, small, with priority 5 finished. Useless sum: 448743748
Child pid 22, small, with priority 5 finished. Useless sum: 448743748
Child pid 42, small, with priority 6 finished. Useless sum: 448743748
Child pid 23, small, with priority 6 finished. Useless sum: 448743748
Child pid 24, small, with priChild pid 43, small, with priority 7 finished. Useless sum: 448743748
ority 7 finished. Useless sum: 448743748
Child pid 44, small, with priority 8 finished. Useless sum: 448743748
Child pid 25, small, with priority 8 finished. Useless sum: 448743748
Child pid 45, small, with priority 9 finished. Useless sum: 448743748
Child pid 26, small, with priority 9 finished. Useless sum: 448743748
Child pid 8, small, with priority 10 finished. Useless sum: 448743748
Child pid 46, small, with priority 10 finished. Useless sum: 448743748
Child pid 27, small, with priority 10 finished. Useless sum: 448743748
Child pid 28, small, with priority 11 finished. Useless sum: 448743748
Child pid 9, small, with priority 11 finished. Useless sum: 448743748
Child pid 47, small, with priority 11 finished. Useless sum: 448743748
Child pid 10, small, with priority 12 finished. Useless sum: 448743748
Child pid 29, small, with priority 12 finished. Useless sum: 448743748
Child pid 30, small, with priority 13 finished. Useless sum: 448743748
Child pid 11, small, with priority 13 finished. Useless sum: 448743748
Child pid 31, small, with priority 14 finished. Useless sum: 448743748
Child pid 12, small, with priority 14 finished. Useless sum: 448743748
Child pid 13, small, with priority 15 finished. Useless sum: 448743748
Child pid 32, small, with priority 15 finished. Useless sum: 448743748
Child pid 14, small, with priority 16 finished. Useless sum: 448743748
Child pid 33, small, with priority 16 finished. Useless sum: 448743748
Child pid 5, large, with priority 16 finished. Useless sum: 1818368003
Child pid 15, small, with priority 17 finished. Useless sum: 448743748
Child pid 34, small, with priority 17 finished. Useless sum: 448743748
Child pid 6, large, with priority 17 finished. Useless sum: 1818368003
Child pid 16, small, with priority 18 finished. Useless sum: 448743748
Child pid 35, small, with priority 18 finished. Useless sum: 448743748
Child pid 7, large, with priority 18 finished. Useless sum: 1818368003
Child pid 36, small, with priority 19 finished. Useless sum: 448743748
Child pid 17, small, with priority 19 finished. Useless sum: 448743748
Child pid 18, small, with priority 20 finished. Useless sum: 448743748
Child pid 37, small, with priority 20 finished. Useless sum: 448743748
Child pid 4, large, with priority 20 finished. Useless sum: 1818368003
```

3) ps (διαδοχικά, μετά την εκτέλεση priotest):