Operating Systems – Assignment 2 Report

Introduction

On this assignment we:

- Learn different process scheduling algorithms.
- Implement different process scheduling algorithms in C programming language.

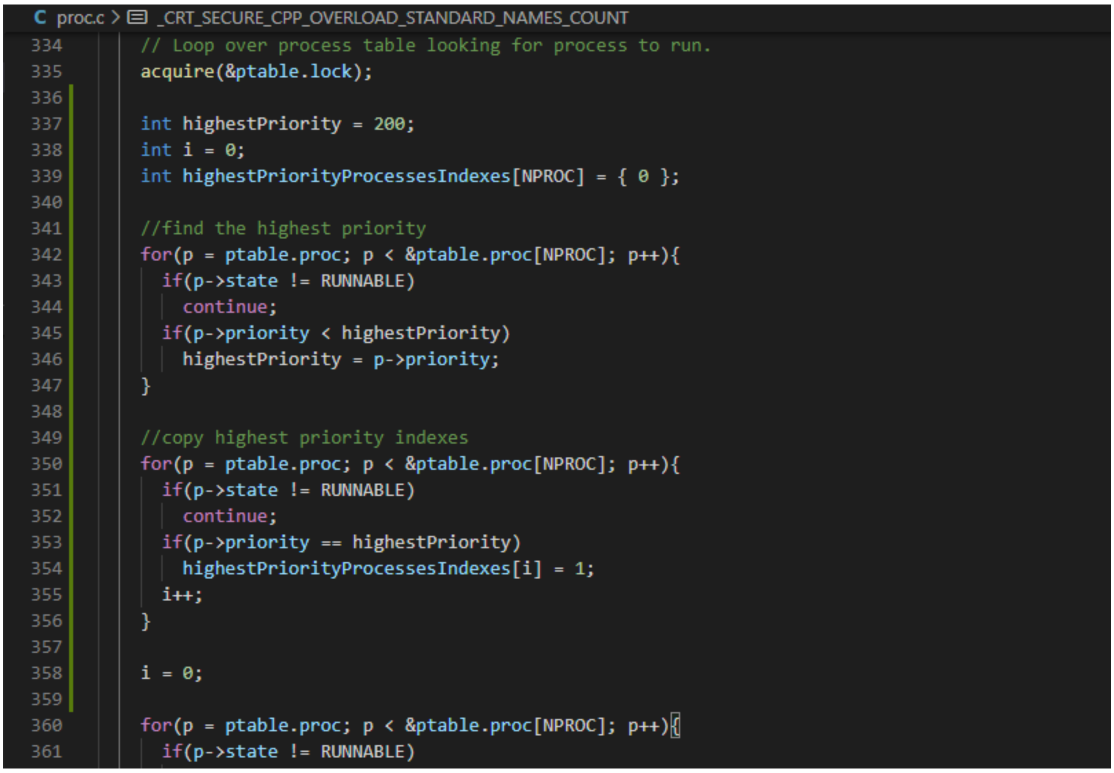
C Program For Implementing Different Scheduling Algorithms

Every function and variable is explained in-code.

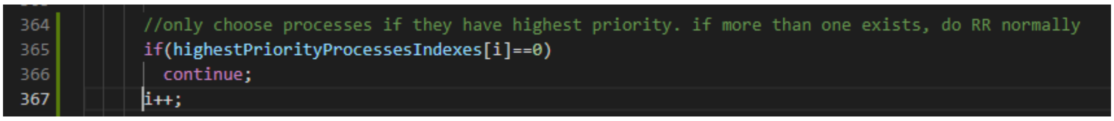
XV6 Scheduling

Implementation Code:

Changes in proc.c(priority implementing):

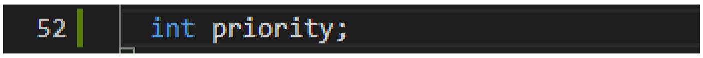


```
// Loop over process table looking for process to run.
acquire(&ptable.lock);

int highestPriority = 200;
int i = 0;
int highestPriorityProcessesIndexes[NPROC] = { 0 };

//find the highest priority
for(p = ptable.proc; p < &ptable.proc[NPROC]; p++){
  if(p->state != RUNNABLE)
    continue;
  if(p->priority < highestPriority)
    highestPriority = p->priority;
}

//copy highest priority indexes
for(p = ptable.proc; p < &ptable.proc[NPROC]; p++){
  if(p->state != RUNNABLE)
    continue;
  if(p->priority == highestPriority)
    highestPriorityProcessesIndexes[i] = 1;
  i++;
}

i = 0;

for(p = ptable.proc; p < &ptable.proc[NPROC]; p++){
  if(p->state != RUNNABLE)
```



```
//only choose processes if they have highest priority. if more than one exists, do RR normally
if(highestPriorityProcessesIndexes[i]==0)
  continue;
i++;
```

Changes in proc.h(priority implementing):

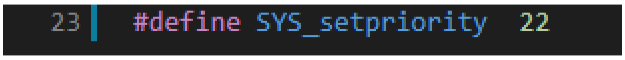


```
int priority;
```

Adding A "setpriority" System Call

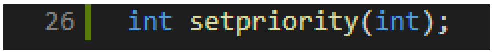
Implementation Code:

Changes in syscall.h:



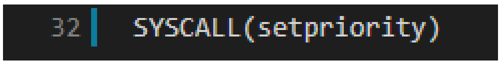
```
#define SYS_setpriority  22
```

Changes in user.h:



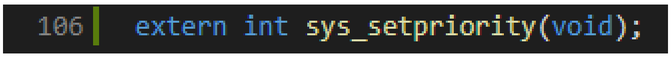
```
int setpriority(int);
```

Changes in usys.s:

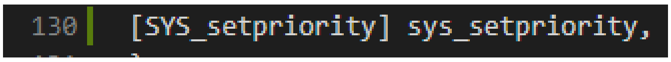


```
SYSCALL(setpriority)
```

Changes in syscall.s:

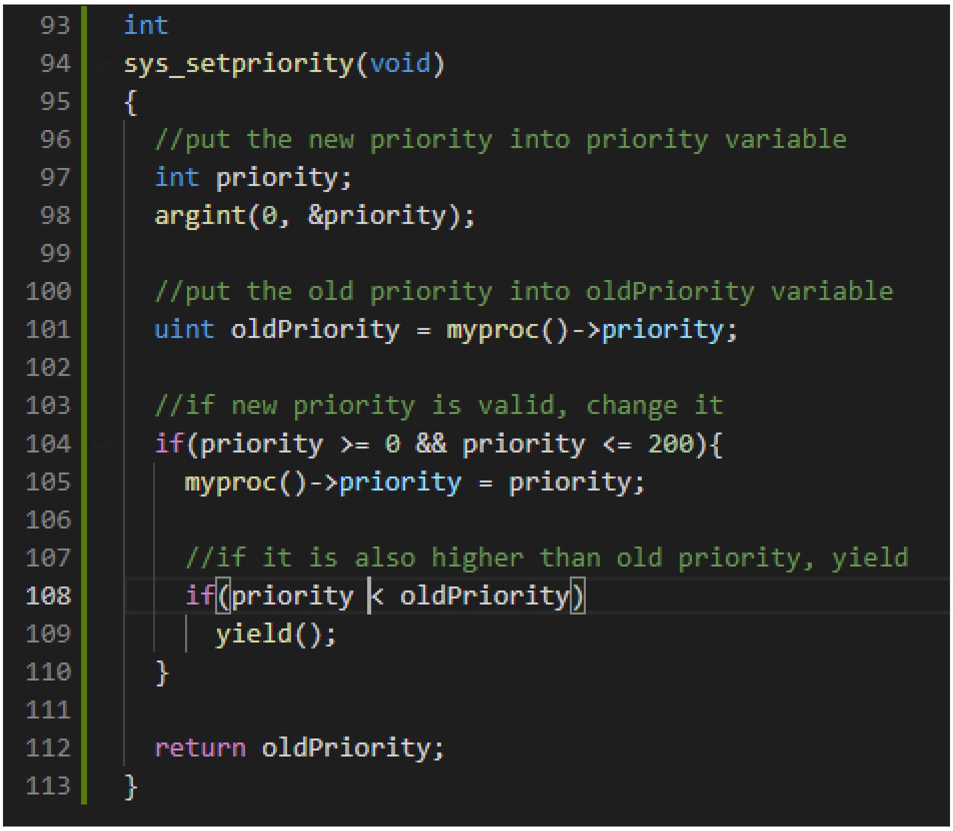


```
extern int sys_setpriority(void);
```



```
[SYS_setpriority] sys_setpriority,
```

Changes in sysproc.c:



```
int
sys_setpriority(void)
{
  //put the new priority into priority variable
  int priority;
  argint(0, &priority);

  //put the old priority into oldPriority variable
  uint oldPriority = myproc()->priority;

  //if new priority is valid, change it
  if(priority >= 0 && priority <= 200){
    myproc()->priority = priority;

    //if it is also higher than old priority, yield
    if(priority < oldPriority)
      yield();
  }

  return oldPriority;
}
```