Project Report

Team members:

1. **Srikanth Goud Nagelli (Nuid:001613157)**
2. **Pranay d.l.v.n.s.k (Nuid:001613816)**

Implemented priority based scheduler and Multithread support in xv6

My role was implementing multithreading support to XV6

1. Multithreading support in xv6:

Added pthread library support along with locks to xv6 i.e implemented functions such as thread_create ,thread join which are similar to pthread_create ,pthread_join
Modified files:syscall.c ,syscall.h ,proc.c,proc.h ,sysproc.c,defs.h,pthread.c ,pthread.h,user.h
System calls Implemented for threads support:
- **Clone:** creates new process similar to fork but allows child process to share part of its execution context with the calling process such as memory space, file descriptors, signal handlers.
- **Join:** waits till all the process reach the zombie state and then it kills all such process

Pthread library functions created for multithreading support:
thread_create,thread_join,lock_acquire,lock_release,lock_init
Application used to test multithreading: mul1
Here five threads are created and multithreading functionality is tested by increasing the globally shared counter independently.

```
$ mul1
thread 0: begins
100000
thread  0: ends
thread 1: begins
200000
thread  1: ends
thread 2: begins
300000
thread  2: ends
thread 3: begins
400000
thread  3: ends
thread 4: begins
500000
thread  4: ends
thread 5: begins
600000
thread  5: ends
```

2. Priority based scheduler implementation in xv6:

Introduction:
This priority based scheduling policy has been implemented in xv6 by modifying the following files proc.c, proc.h, sysproc.c, sysproc.h, syscall.c, syscall.h, user.h, usys.S, defs.h
Three additional system calls has been added to xv6 for the implementation they are

- Schedgraph-this system call sets the flag to reperesent the selection of process graphically.

*Definition:*void schedgraph(int);

- Sched_type-this system call allows user to select the type of the scheduler (either RR or Priority)

*Definition:*int sched_type(int);

- Setpriority-this system call allows the user to assign priority for each process according to the pid.

*Definition:*void setpriority(int,int);

Working:
To test the working of scheduler we have written an application program(testc.c) that creates NCHILDS(we created 8 childs for testing purpose) specified in the header and these child processes communicates with the parent by using pipe, the procedure of choosing a particular child process to run tests the working of scheduler.
The application is used as(after make qemu):

```
$ testc
Usage: testc [0:RoundRobin or 1:priority based scheduler]
```

testc 0-RoundRobin scheduler(default) | testc 1-Priority based scheduler

Round Robin scheduler:

```
$ testc 0

Using round robin(default) scheduler
scheduler selected:RoundRobin is
Process[pid=13]
Process[pid=14]
Process[pid=15]
Process[pid=16]
Process[pid=17]
Process[pid=18]
Process[pid=19]
Process[pid=20]
```

Round robin scheduler,

Created 8 process with equal priority (default 2)

pid (13-20)

Priority based scheduler:

```
$ testc 1

Using the priority based scheduler
scheduler selected:Priority based scheduler is
Process[pid=22 priority=1]
Process[pid=23 priority=1]
Process[pid=24 priority=2]
Process[pid=25 priority=2]
Process[pid=26 priority=3]
Process[pid=27 priority=3]
Process[pid=28 priority=4]
Process[pid=29 priority=4]
```

Priority based scheduler,

Process pid 22,23-priority=1

Process pid 24,25-priority=2

Process pid 26,27-priority=3

Process pid 28,29-priority=4

Scheduling graph(Round Robin scheduler):

```
13 (2)| 14 (2)| 15 (2)| 16 (2)| 17 (2)| 18 (2)| 19 (2)| 20 (2)| 13 (2)| 14 (2)|
15 (2)| 16 (2)| 17 (2)| 18 (2)| 19 (2)| 20 (2)| 13 (2)| 14 (2)| 15 (2)| 16 (2)|
17 (2)| 18 (2)| 19 (2)| 20 (2)| 13 (2)| 14 (2)| 15 (2)| 16 (2)| 17 (2)| 18 (2)|
19 (2)| 20 (2)| 13 (2)| 14 (2)| 15 (2)| 16 (2)| 17 (2)| 18 (2)| 19 (2)| 20 (2)|
13 (2)| 14 (2)| 15 (2)| 16 (2)| 17 (2)| 18 (2)| 19 (2)| 20 (2)| 13 (2)| 14 (2)|
15 (2)| 16 (2)| 17 (2)| 18 (2)| 19 (2)| 20 (2)| 13 (2)| 14 (2)| 15 (2)| 16 (2)|
17 (2)| 18 (2)| 19 (2)| 20 (2)| 13 (2)| 14 (2)| 15 (2)| 16 (2)| 17 (2)| 18 (2)|
19 (2)| 20 (2)| 13 (2)| 14 (2)| 15 (2)| 16 (2)| 17 (2)| 18 (2)| 19 (2)| 20 (2)|
13 (2)| 14 (2)| 15 (2)| 16 (2)| 17 (2)| 18 (2)| 19 (2)| 20 (2)| 13 (2)| 14 (2)|
15 (2)| 16 (2)| 17 (2)| 18 (2)| 19 (2)| 20 (2)| 13 (2)| 14 (2)| 15 (2)| 16 (2)|
17 (2)| 18 (2)| 19 (2)| 20 (2)| 13 (2)| 14 (2)| 15 (2)| 16 (2)| 17 (2)| 18 (2)|
19 (2)| 20 (2)| 13 (2)| 14 (2)| 15 (2)| 16 (2)| 17 (2)| 18 (2)| 19 (2)| 20 (2)|
13 (2)| 14 (2)| 15 (2)| 16 (2)| 17 (2)| 18 (2)| 19 (2)| 20 (2)| 13 (2)| 14 (2)|
15 (2)| 16 (2)| 17 (2)| 18 (2)| 19 (2)| 20 (2)| 13 (2)| 14 (2)| 15 (2)| 16 (2)|
17 (2)| 18 (2)| 19 (2)| 20 (2)| 13 (2)| 14 (2)| 15 (2)| 16 (2)| 17 (2)| 18 (2)|
19 (2)| 20 (2)| 13 (2)| 14 (2)| 15 (2)| 16 (2)| 17 (2)| 18 (2)| 19 (2)| 20 (2)|
13 (2)| 14 (2)| 12 (2)| 18 (2)| 14 (2)| 18 (2)|
```

pid(priority)

Scheduling graph(Priority based scheduler):

```
23 (1)| 22 (1)| 23 (1)| 22 (1)| 23 (1)| 22 (1)| 23 (1)| 22 (1)| 23 (1)| 22 (1)|
23 (1)| 22 (1)| 23 (1)| 22 (1)| 23 (1)| 22 (1)| 23 (1)| 22 (1)| 23 (1)| 22 (1)|
23 (1)| 22 (1)| 23 (1)| 22 (1)| 23 (1)| 22 (1)| 23 (1)| 22 (1)| 23 (1)| 22 (1)|
23 (1)| 22 (1)| 23 (1)| 22 (1)| 23 (1)| 22 (1)| 23 (1)| 22 (1)| 23 (1)| 24 (2)|
25 (2)| 24 (2)| 25 (2)| 24 (2)| 25 (2)| 24 (2)| 25 (2)| 24 (2)| 25 (2)| 24 (2)|
25 (2)| 24 (2)| 25 (2)| 24 (2)| 25 (2)| 24 (2)| 25 (2)| 24 (2)| 25 (2)| 24 (2)|
25 (2)| 24 (2)| 25 (2)| 24 (2)| 25 (2)| 24 (2)| 25 (2)| 24 (2)| 25 (2)| 24 (2)|
25 (2)| 24 (2)| 25 (2)| 24 (2)| 25 (2)| 24 (2)| 25 (2)| 24 (2)| 25 (2)| 26 (3)|
27 (3)| 26 (3)| 27 (3)| 26 (3)| 27 (3)| 26 (3)| 27 (3)| 26 (3)| 27 (3)| 26 (3)|
27 (3)| 26 (3)| 27 (3)| 26 (3)| 27 (3)| 26 (3)| 27 (3)| 26 (3)| 27 (3)| 26 (3)|
27 (3)| 26 (3)| 27 (3)| 26 (3)| 27 (3)| 26 (3)| 27 (3)| 26 (3)| 27 (3)| 26 (3)|
27 (3)| 26 (3)| 27 (3)| 26 (3)| 27 (3)| 26 (3)| 27 (3)| 26 (3)| 29 (4)| 28 (4)|
29 (4)| 28 (4)| 29 (4)| 28 (4)| 29 (4)| 28 (4)| 29 (4)| 28 (4)| 29 (4)| 28 (4)|
29 (4)| 28 (4)| 29 (4)| 28 (4)| 29 (4)| 28 (4)| 29 (4)| 28 (4)| 29 (4)| 28 (4)|
29 (4)| 28 (4)| 29 (4)| 28 (4)| 29 (4)| 28 (4)| 29 (4)| 28 (4)| 29 (4)| 28 (4)|
29 (4)| 28 (4)| 29 (4)| 28 (4)| 29 (4)| 21 (2)| 28 (4)|
```

pid(priority)