OSN Assignment Report

K. L. Nirmala (2021101126)
V. Siva Koti Reddy (2021101135)

Specification 1:

Strace:

The files to be added are as follows:

- user/strace.c - contains code for strace.

For adding a system call we need to change the following files.

- kernel/sysproc.c
- kernel/syscall.h
- kernel/syscall.c
- user/user.h
- user/usys.pl
- MAKEFILE

Sigalarm and Sigreturn:

- Added variables in kernel/proc.h file
- Defined syscalls sigalarm, sigreturn in kernel/syscall.h
- Added extern prototypes in kernel/syscall.c and then implemented
- Added following commands to handle interrupts in usertrap function in trap.c file

```
if (which_dev == 2)
  {
    if (p->alarm != 0)
    {
      p->duration++;
      if (p->duration == p->alarm)
      {
        p->duration = 0;
        if (p->alarm_trapframe == 0)
        {
          p->alarm_trapframe = kalloc();
          memmove(p->alarm_trapframe, p->trapframe, 512);
          p->trapframe->epc = p->handler;
        }
```

```
      else
      {
        yield();
      }
    }
    else
    {
      yield();
    }
  }
  else
  {
    yield();
  }
}
```

- Defined two functions in user/user.h file

```
int sigalarm(int ticks, void (*handler)());
int sigreturn(void);
```

Specification 2:

Scheduling:

We have implemented the following scheduling algorithms:

- First Come First Serve (FCFS)
- Priority Based Scheduling (PBS)
- Lottery Based Scheduling (RANDOM)
- Multi Level Feedback Queue (MLFQ)

1. FCFS:

- CPU is allocated to the processes in the increasing order of the creation times.
- We implemented the code in kernel/proc.c in scheduler function in a way allocating CPU to processes having less creationtime (a variable) first.

2. PBS:

We add new variables in struct proc called

- ❖ runTime - the run time of the process
- ❖ sleepTime - the sleep time of the process
- ❖ staticPriority - the priority of the process
- ❖ numscheduled - the number of times a process has been allocated CPU
- ❖ niceness - the niceness of the process

+++

- ❖ Added a system call set_priority that is used to update the priority of the process.

- ❖ The meaning of niceness values are: 5 is neutral, 10 helps priority by 5, 0 hurts priority by 5.

- ❖ The scheduler function traverses and allocates the CPU to the process that has the highest dynamic priority.

3. Lottery Based:

- We generate a tckt using the random generator and we check for the process which has the tckt and allocate the CPU to that process.

4. MLFQ:

There will be 5 queues denoting the set of process executed in the CPU.

The nth queue processes start to execute if and only if the all processes in the above n-1 queues are done executing. This is the key idea of MLFQ scheduling algorithm.

Scheduling Analysis:

From schedulertest.c

Scheduler	FCFS	PBS	Lottery	MLFQ
Avg. runTime	18	17	18	18
Wait Time	173	172	174	172

Specification 3: Copy-On Write fork:

Whenever we fork a process a child and parent are created and a copy of the page table of the parent is created and allocated to the the child, this increases data redundancy even though it is not needed always.

A copy needs to be created only if the child process not the page table of the parent needs to write in the pagetable.

So in the Copy on Write we check if the child process writes in the pagetable and if not we can allocate the parent and child process a shared memory of the pagetable to read the data. else, we create a copy of the page table of the parent and assign it to the child.

Implementation:

- Implemented uvmcowcopy(), cowcheckpage() functions in vm.c.
- Made changes in trap.c.
- Modified usertrap() in trap.c with else if condition on r_scause()
- Declared and implemented the functions, krefpage(), kcopy_n_deref() in kalloc.c

Cow-test:

- Added cowtest.c in user for testing the syscall.
- Made necessary changes in makefile.