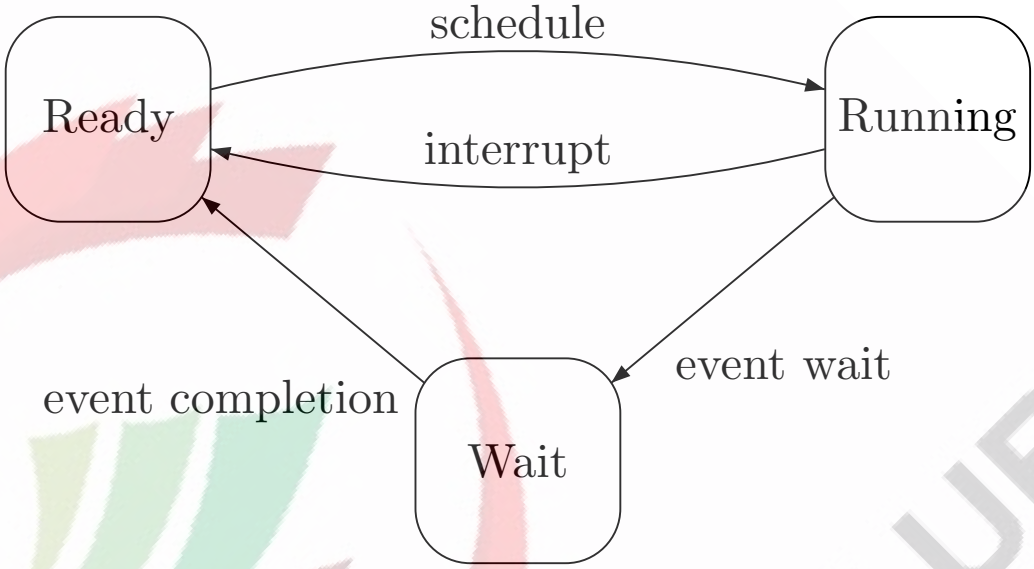
Notre Dame University
CSC 414 Applied Operating Systems
Exam 1
Spring 2009

Duration: 1 Hour
Name:
ID:

Instructions

1. Write your answers in the space provided on the question sheet.
2. Use the back pages for rough work.
3. There are a total of 9 pages containing 9 questions. Two pages are blank (pages 4 and 8).
4. The last page contains some information about the Linux scheduler.
5. **There is NO correlation between the space provided for each problem and the space needed. If you need more space write on the back and clearly indicate that you did so.**

1. (5pts) Draw the state transition diagram containing "running", "ready" and "waiting" states only. List the events that cause each transition.



2. (5pts) What is a context switch and how it is handled by the OS? Where is the context of a process saved?
   A context switch is the act of saving the context of the current process and loading the context of the next process to run. The context of process is saved in the PCB of the process in the memory address space of the OS

3. (5pts) Describe how the system goes from user to system privilege.
   The system switches automatically to system privilege when an interrupt occurs.

4. (5pts) What are the modes of operations of a CPU? how many are usually needed by an OS?
   these are the privilege levels. Usually the OS needs two: a user and OS mode

5. (5pts) Explain how a user application can perform (directly or indirectly) a privileged operations?
   Via system calls which are implemented by interrupts which means the CPU switches automatically to system mode

6. (20 pts) grep is a program that filters an input stream according to a pattern passed on the command line. For example if the pattern is "xyz", only the lines in the input containing "xyz" will be printed on the output. Use *fork()*, *execlp* (or equivalent), *dup2*, *pipe()* and *open* to execute the shell command "sort -r <input |grep xyz > output" where "input" and "output" are the names of files. Note that your code should execute this specific command only (no parsing).

Solution

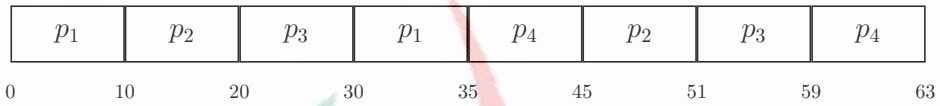
```
int main(){
 int p[2],fd;
 pid_t pid;

 pipe(p);
 pid=fork();
 if(pid==0){
   fd=open("input",O_RDONLY,0);
   dup2(fd,0);
   dup2(p[1],1);
   close(fd);close(p[0]);close(p[1])
   execlp("sort","sort","-r",0);
  }
 else{
   fd=open("output",O_WRONLY,0);
    dup2(fd,1);
    dup2(p[0],0);
    close(fd);close(p[0]);close(p[1]);
    execlp("grep","grep","xyz",0);
 }
}
```

7. (20pts) Four Processes p_1,p_2,p_3 and p_4 arrive in the ready queue at times 0,3,8,11 and their running time are 15,16,18 and 14 clock ticks respectively. Draw the Gantt chart for the system and compute the average waiting and response time for each of the following algorithms. Ignore process switching time

   (a) Round Robin (quantum=10).

   (b) Shortest remaining time first (i.e. preemptive SJF).

Round Robin



SRF

1

8. (15pts)A system uses round robin scheduling with a quantum of Q seconds. Assume that we have n processes and each runs T seconds and that process switching takes S seconds. Since process switching is an overhead compute the average efficiency(ratio of useful time divided by total time) of the CPU for the following cases

   (a) $T = 2Q$

   (b) $Q = 2T$

   (c) $Q = S < T$

Solution

When the quantum $Q > T$ then on average a process can finish before its quantum expires and there will be one context switch. In this case the efficiency is $\frac{T}{T+S}$. When $Q < T$ a process needs T/Q context switches to finish and therefore the overhead is ST/Q and the efficiency is $\frac{T}{T+ST/Q}$

   (a) $\frac{Q}{Q+S}$

   (b) $\frac{T}{T+S}$

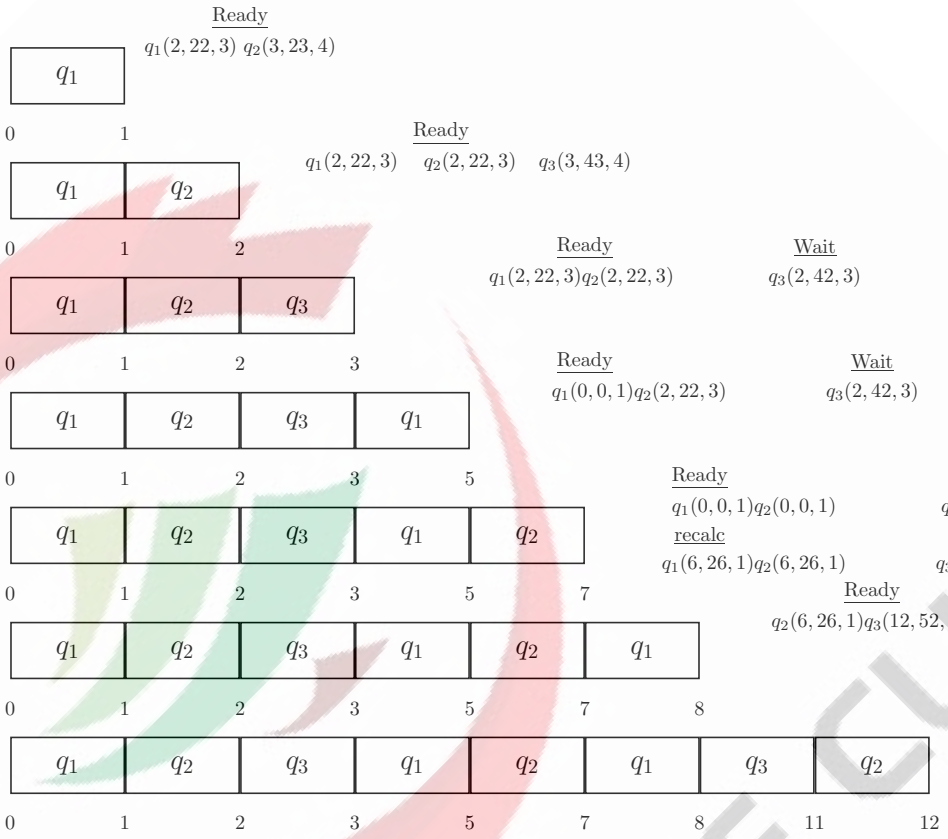
   (c) $\frac{1}{2}$

9 (20pts) Three processes q_1 with nice value 0, q_2 with nice value 0 and q_3 with nice value -20, are scheduled on a Linux system. q_1 starts at t=0 and later on forks q_2 and q_2 forks q_3 as shown below. q_1 writes to a buffer shared with q_3. q_3 issues a blocking read on the shared buffer. Assume that for all processes the initial counter value is 3. (Note that *fork()*, *write()* and *read()* are system calls).

(a) Draw the Gantt chart for the system.

(b) What is the total number of context switches.

q_1		q_2		$q3$	
inst	ticks	inst	ticks	inst	ticks
fork()	1	fork()	1	read()	1
do work	2	do work	3	do work	3
write()	1				



1