**SCHOOL OF COMPUTING AND IT**

**II B.Tech. IV Semester; First- Sessional Examination, Feb-2017**

**Branch: CSE / IT /CCE**

**(SOLUTION)**

**Subject Code: CS1401**

**Subject Name : Operating Systems**

**Max. Marks: 20 Duration: 1 hour**

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Instructions:**

* All questions are compulsory
* Missing data if any can be suitably assumed
* Two books and one notebook / spiral bound book is allowed.

1. Does context switching takes place whenever there is a mode switching? Explain by giving examples of both hardware and software interrupts. [3]

Mode switching is different from context switching. Whenever software or hardware interrupt is there, CPU mode changes from ordinary to privileged mode and control is transferred to operating system kernel. After that, if control is returned to the same user space process which was running prior to interrupt, then context switching will not take place.

1. Write a program in ‘C’ (Linux OS) to create a process tree as under and displays process identifier of each process:

`

[3]

#include <stdio.h>

#include <unistd.h>

#include <sys/types.h>

void main()

{

pid\_t pid,pid1,pid2;

pid=fork();

if (pid==0)

{

pid1=fork();

if (pid1>0)

{

wait(NULL);

printf("Process Id: Child#2 = %d\tParent = %d\n",getpid(),getppid());

pid2=fork();

if (pid2==0)

printf("Process Id: Child#4 = %d\tParent = %d\n",getpid(),getppid());

}

else

if (pid1==0)

{

printf("Process Id: Child#3 = %d\tParent = %d\n",getpid(),getppid());

}

}

else

{

wait(NULL);

printf("Process Id: Child#1 = %d\tParent = %d\n",getpid(),getppid());

}

}

1. Write a program in ‘C’ (Linux OS) which displays numbers from 1 to 10 and alphabets from A to E alternately ( 1 2 A 3 4 B …. ). The program should have two threads (pthreads), One of the thread displays numbers and other one displays alphabets. (Hint : Use sleep()) [4]

#include <stdio.h>

#include <pthread.h>

void \*numbers(void \*param)

{

int i;

for(i=1;i<=10;i++)

{

printf("%d ",i);

sleep(1);

}

pthread\_exit(0);

}

void \*alpha(void \*param)

{

int i;

for(i=65;i<=69;i++)

{

sleep(2);

printf("%c ",i);

}

pthread\_exit(0);

}

void main(int c,char \*argv[])

{

pthread\_t tid1,tid2;

pthread\_attr\_t attr;

int x,y;

pthread\_attr\_init(&attr);

pthread\_create(&tid1,&attr,numbers,NULL);

pthread\_create(&tid2,&attr,alpha,NULL);

pthread\_join(tid1,NULL);

pthread\_join(tid2,NULL);

//printf("sum is %d",sum);

}

1. Write a program in ‘C’ (Linux OS) in which the child process sends the name of one of the existing file on disk to the parent process and the parent process displays the contents of that file. The communication between the parent and the child is through pipe. [4]

#include <stdio.h>

#include <unistd.h>

#include <sys/types.h>

#include <fcntl.h>

main()

{

int pfd[2], id, n, fd;

char buff[512], buffer [512];

pipe(pfd);

id=fork();

if (id==0)

{

close (pfd[0]);

//taking filename from user

printf("Enter File Name \n");

scanf("%s",buff);

write (pfd[1], buff,n);

}

else

{

close (pfd[1]);

//reading file name from pipe

n=read (pfd[0],buff,512);

fd = open(buff, O\_RDONLY);

while ((n=read(fd,buffer,512))>0)

write (1,buffer,n);

}

}

1. Consider the following set of processes, with the length of the CPU-burst time given in milliseconds (Higher priority number means lower priority ):

|  |  |  |  |
| --- | --- | --- | --- |
| Process | Arrival Time | CPU Burst Time | Priority |
| P1 | 0 | 12 | 3 |
| P2 | 1 | 10 | 4 |
| P3 | 3 | 5 | 2 |
| P4 | 5 | 2 | 1 |

1. Draw four Gantt charts illustrating the execution of these processes using FCFS, SJF (Preemptive), Priority (Preemptive) and RR (quantum = 2) scheduling.
2. What is the turnaround time of each processes for Priority (Preemptive) and RR scheduling algorithms?
3. What is the waiting time of each processes for Priority (Preemptive) and RR scheduling algorithms? [2 +2 +2 = 6]

FCFS

|  |  |  |  |
| --- | --- | --- | --- |
| P1 | P2 | P3 | P4 |
| 0 - 12 | 12-22 | 22-27 | 27-29 |

SJF (Pre-emptive)

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Bal Time | 11 | 8 | 3 | 0 | 0 | 0 | 0 |
| Proc. No. | P1 | P2 | P3 | P4 | P3 | P2 | P1 |
| Time | 0-1 | 1-3 | 3-5 | 5-7 | 7-10 | 10-18 | 18-29 |

Priority (Pre-emptive)

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Bal Time | 9 | 3 | 0 | 0 | 0 | 0 |
| Proc. No. | P1 | P3 | P4 | P3 | P1 | P2 |
| Time | 0-3 | 3-5 | 5-7 | 7-10 | 10-19 | 19-29 |

RR (Q = 2)

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Bal Time | 10 | 8 | 8 | 3 | 6 | 0 | 6 | 1 | 4 | 4 | 0 | 2 | 2 | 0 | 0 |
| Proc. No. | P1 | P2 | P1 | P3 | P2 | P4 | P1 | P3 | P2 | P1 | P3 | P2 | P1 | P2 | P1 |
| Time | 0-2 | 2-4 | 4-6 | 6-8 | 8-10 | 10-12 | 12-14 | 14-16 | 16-18 | 18-20 | 20-21 | 21-23 | 23-25 | 25-27 | 27-29 |

|  |  |
| --- | --- |
| Priority (Pre-emp) | RR (Q = 2) |

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Proc. | AT | BT | Prio. | CT | TAT | WT | CT | TAT | WT |
| P1 | 0 | 12 | 3 | 19 | 19 | 7 | 29 | 29 | 17 |
| P2 | 1 | 10 | 4(L) | 29 | 28 | 18 | 27 | 26 | 16 |
| P3 | 3 | 5 | 2 | 10 | 7 | 2 | 21 | 18 | 13 |
| P4 | 5 | 2 | 1(H) | 7 | 2 | 0 | 12 | 7 | 5 |

56 27 80 51

Av 14 6.75 20 12.75