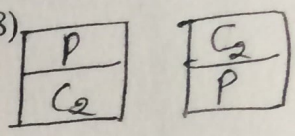
27 b

d) What are the possible sequence of execution of the last printf() and also specify the values of n and m printed? (4*2=8)

Ans: Two possible sequence of execution :→



① If Parent execute before C2 :→

Last printf	46	67	(due to P)
Last print f	41	62	(due to C2)

② If child(C2) execute first (before parent) :→

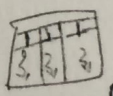
Last print f	41	62	(due to C2)
Last print f	46	67.	(due to P)

4. What happens when a child completes its execution before the parent executes a wait() system calls? What happens when parent dies before the child finishes its execution.(assume Linux /UNIX OS) (2)

Ans: ① In this case the execution will not affect the functioni in any way. As wait system call ensures that atleast one of the children should complete execution before parent starts execution ; and this condition is already fulfilled. Hence no waiting for parent.

② In this case ; the child becomes orphan. And in Linux/UNIX OS ; the orphan child processes have 'init' process as its parent. So child executes its part of code and when exit() ; returns its process id to init process.

5. Can a multithreaded solution using multiple user level threads achieve better performance on a multiprocessor system than on a single processor system?(2)



3,2,2

Ans: On a single processor system, execution will be normal ; But on a multiprocessor system ; OS have certain options for execution like should all the process / certain process / only one process which is having the operation (any specified operation) should respond. Then On Multiprocessor ; it is efficient and can achieve better performance.

6. Given the following job mix

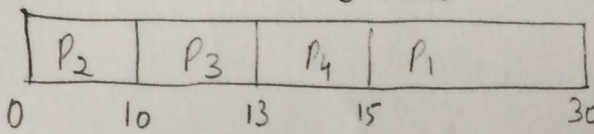
Process	Arrival time	Burst time (in msecs.)	Priority
P1	0	15	6
P2	0	10	4
P3	3	3	0 (Highest)
P4	7	2	2

Calculate the average turn around time and average waiting time for:

WT = waiting time

a) A non-preemptive priority algorithms

Ans:



avg waiting time = $\frac{(WT)_{P_1} + (WT)_{P_2} + (WT)_{P_3} + (WT)_4}{4}$

$= \frac{(15-0) + (0-0) + (10-3) + (13-7)}{4}$

$= \frac{28}{4} = 7$ msec

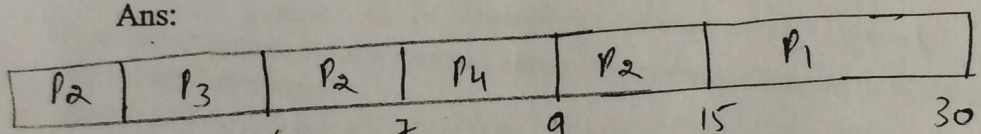
avg turnaround time = (CPU time + I/O time + waiting time (due to I/O device queue) + waiting time (due to ready queue))

$= \frac{(15+15) + (10+0) + (3+7) + (2+6)}{4}$

$= \frac{58}{4} = 14.5$ msec

b) A preemptive priority algorithms. (4)

Ans:



avg waiting time = $\frac{(15-0) + (9-3-1-0) + (3-3) + (7-7)}{4}$

$= \frac{15+5}{4} = \frac{20}{4} = 5$ msecs

avg turn around time = avg waiting time + avg CPU burst time

$= 5 + \left(\frac{15+10+3+2}{4}\right) = 5 + \frac{30}{4} = 5+7.5$

$= 12.5$ msecs

actually turn around time = CPU time (burst) + I/O time (burst) + (waiting time due to I/O device queue) + (waiting time due to ready queue);

~~No info about device queue, hence not considered~~

Roll No: B130055CS 3½72 Name ASHOK DHUNGANA

National Institute of Technology Calicut

Department of Computer Science & Engineering

CS3003 – OPERATING SYSTEMS

First MID Term Exam (Monsoon Semester 2015) **Max. Marks:7**

PART B **Time: 30 mnts**

---

1. Which type of fragmentation(s) does the following memory management schemes suffer from?

   Paging : *Ans:* ~~internal~~ external [0.5]

   Segmentation : *Ans:* ~~external~~ internal. [0.5]

2. Consider six memory partitions of size 200 KB, 400 KB, 600 KB, 500 KB, 300 KB, and 250 KB, where KB refers to kilobyte. These partitions need to be allotted to four processes of sizes 357 KB, 210 KB, 468 KB and 491 KB in that order. If the best fit algorithm is used, which partitions are NOT allotted to any process? [1]

   Ans: 200 KB, 300 KB

3. Given an effective access time of 300ns, a main memory access time of 250ns, a TLB access time of 20ns what is the hit rate? hit rate = x [1]

   Ans:

   $300 = (20+250)x + (1-x)(20+250+250)$

   $300 = 270x + (1-x)\,520$

   $300 = -250x + 520$

   $250x = 220 \quad \therefore x = \frac{220}{250} = 0.88$ ∴ hit rate = 88%.

4. Consider the virtual page reference string 7,1,2,4,5,1,2,4,3,7,5,8,3,2,7, 3, 8, 2, 1, 1, 7, 4, 7, 2, 8, 4, 7, 1, 5. On a demand paged virtual memory system running on a computer system that main memory size of 4 pages frames which are initially empty. Let LRU, FIFO and OPTIMAL denote the number of page faults under the corresponding page replacements policy. Then choose the right relation and How it comes [2]

   a. OPTIMAL < LRU < FIFO

   b. OPTIMAL < FIFO < LRU ✓ (b)

   c. OPTIMAL = LRU

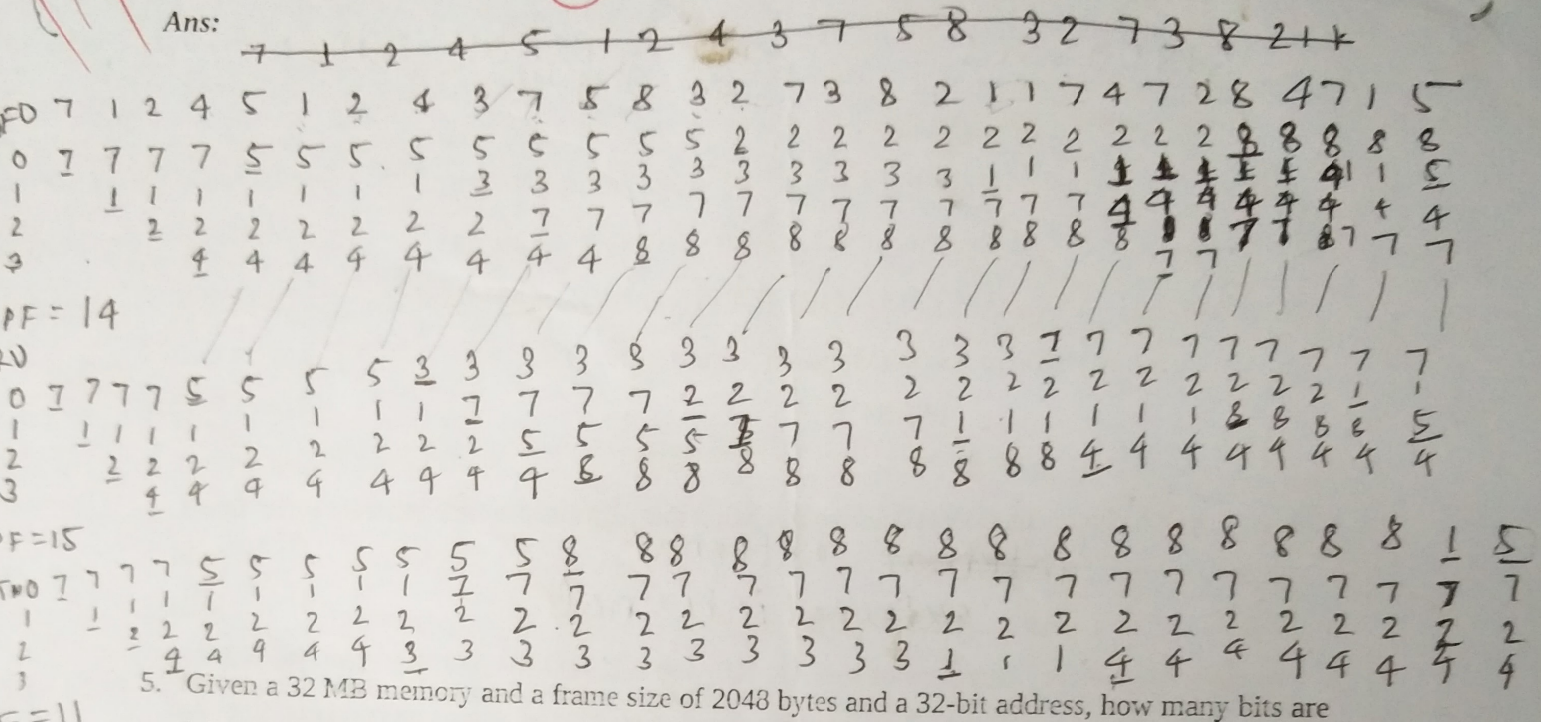
   d. OPTIMAL = FIFO

FIFO PF = 14

LRU PF = 15

OPTM PF = 11

Ans:



5. Given a 32 MB memory and a frame size of 2048 bytes and a 32-bit address, how many bits are used for the page offset and how many bits are used to index into a page table? Please decode the following address in terms of page number and offset within the page: 00000000000000001100100000001011. [2]

Ans:

No. of frames = $\frac{32 \times 2^{20}}{2 \times 2^{10}} = 16 \times 2^{10} = 2^4 \times 2^{10} = 2^{14}$

No. of page table entries = $2^{32}/2 \times 2^{10} = 2^{21}$

∴ 21 bits are used to ~~index into page table~~

11 ~~bits~~ are used for page offset.

∴ 21 bits are used to index into page table

11 bits are used for page offset.

Given address

00000000000001100101 00000001011

21 bit 11 bit

page number = 25

page offset = 11

Roll No: B130057CS Name PIYUSH BHOPALKA

5½/4

National Institute of Technology Calicut

Department of Computer Science & Engineering

CS3003 – OPERATING SYSTEMS

First MID Term Exam (Monsoon Semester 2015) **Max. Marks: 8**

PART A **Time: 30mnts**

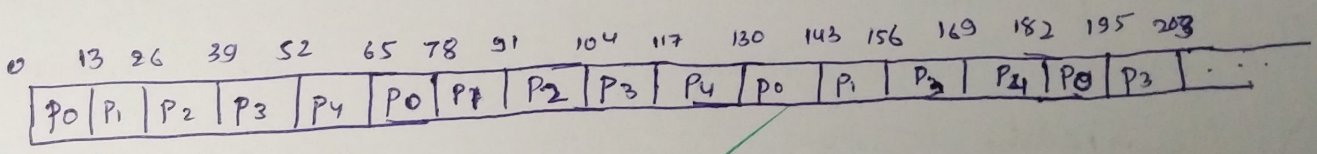
---

1. Assume you have following jobs to execute with one processor. Suppose system uses RR Scheduling with time slice as 10 units. Assume context switch time is 3 units.

Pi	Service time(Pi)
0	85
1	30
2	20
3	35
4	45

a) What is the turnaround time for the process P3? [1 mark]

Ans: Turnaround time = 200 units ✓



b) What is the ~~average~~ wait time of P3? [1 mark]

Ans: 39 units

c) What is the effect of increasing the time quantum to an arbitrarily large number for Round Robin scheduling? [0.5 mark]

Ans: The turnaround time and the wait time for the processes will increase.

2. Linux 2.6.23 kernel uses .....Scheduler which uses ..... data structure as run queue whose run level complexity for scheduling is.....O(n)..... [1.5 marks]

Fair-Share Scheduler doubly linked list

3. System calls are function calls made from user space programs into the kernel requesting for some services. How will the system passes system call number from user space to kernel space? [0.5 Mark]

Ans: With the help of registers

4. Fork returns ...parent process id..... to parent process on success [0.5 mark]

Ans:

5. How many times printf() will be executed in the below mentioned program?

```
main() {
    int i;
    for (i = 0; i < 4; i++)
    fork();
    printf("my pid = %d\n", getpid());
}
```

[1 Mark]

Ans: 16 times

6. What are the process states in Linux [1 Mark]

Ans: ready to run
sleep

7. A ...orphan..... is a process that is still executing, but whose parent has died.

Ans: [0.5 Mark]

8. A process which has finished its execution, but its parent has not yet called to collect its return status, becomes a...zombie..... [0.5 Mark]

Ans:

Roll No: B130056CS Name Prabhav Adhikari

National Institute of Technology Calicut

Department of Computer Science & Engineering

CS3003 – OPERATING SYSTEMS

First MID Term Exam (Monsoon Semester 2015) **Max. Marks:7**

PART B **Time: 30 mnts**

1. Which type of fragmentation(s) does the following memory management schemes suffer from?

   Paging : *Ans:* external [0.5]

   Segmentation : *Ans:* internal & external [0.5]

2. Consider six memory partitions of size 200 KB, 400 KB, 600 KB, 500 KB, 300 KB, and 250 KB, where KB refers to kilobyte. These partitions need to be allotted to four processes of sizes 357 KB, 210 KB, 468 KB and 491 KB in that order. If the best fit algorithm is used, which partitions are NOT allotted to any process? [1]

   Ans: For 357 KB → 400 KB
   210 KB → 250 KB
   468 KB → 500 KB
   491 KB → 600 KB

   So, 200 KB, 300 KB won't be allotted to any process.

3. Given an effective access time of 300ns, a main memory access time of 250ns, a TLB access time of 20ns what is the hit rate? [1]

   Ans: Effective access time = (hit ratio) * TLB access time + (1 - hit ratio) * (Main Memory access time + TLB)

   $300 = \alpha * 20 + (1-\alpha)(250+20) \Rightarrow 300 = 20\alpha + 270 - 270\alpha$

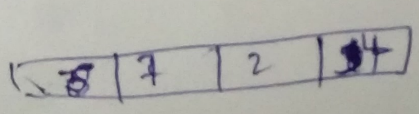
   $\Rightarrow 30 = -250\alpha \Rightarrow \alpha = -0.12$

4. Consider the virtual page reference string 7,1,2,4,5,1,2,4,3,7,5,8,3,2,7, 3, 8, 2, 1, 1, 7, 4, 7, 2, 8, 4, 7, 1 5. On a demand paged virtual memory system running on a computer system that main memory size of 4 pages frames which are initially empty. Let LRU, FIFO and OPTIMAL denote the number of page faults under the corresponding page replacements policy. Then choose the right relation and How it comes [2]

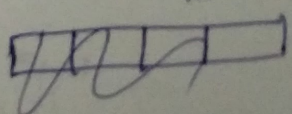
   a. OPTIMAL < LRU < FIFO
   b. OPTIMAL < FIFO < LRU (OPTIMAL = 12, FIFO = 16, LRU = 17)
   c. OPTIMAL = LRU
   d. OPTIMAL = FIFO

7 1 2 4 5 1 2 4 3 7 5 8 3 2 7 3 8 2 1 1 7 4 7 2 8 4 7 1 5

Strategy for optimal: least farthest in future



OPTIMAL = 12



LRU.

4 + 2 + 1 + 1 + 1 + 1 + 2

Ans: LRU = 14+2 = 16 | LRU = 17

FIFO = 13

FIFO = 16

5. Given a 32 MB memory and a frame size of 2048 bytes and a 32-bit address, how many bits are used for the page offset and how many bits are used to index into a page table? Please decode the following address in terms of page number and offset within the page: 00000000000000001100100000001011. [2]

Ans:

Frame size = 2^{11} bytes.

Address = 2^{32} bytes = 2^{25} bytes

So, Total pages = $\frac{2^{25}}{2^{11}} = 2^{14}$

$\frac{2^{25}}{2^{11}} = 2^{14}$.

14 bits needed to index into page table.

11 bits needed for offset.

Address = $2^{15} + 2^{14} + 2^{11} + 2^3 + 2^1 + 2^0 = 51211$ bytes.

Page No. = $\frac{51211}{2^{14}} = 3$

offset = 51211 % 2^{14}

offset = 2059

Page = 3

offset = 2059

Department of Computer Science & Engineering

Roll No.....

NATIONAL INSTITUTE OF TECHNOLOGY CALICUT
Monsoon Semester 2015
End Semester Examination
CS 3003 OPERATING SYSTEMS

Duration: 1hour30mnts **PART B** **Maximum Marks: [25]**

1. Name the following:

   a. Frequent page faults results in an undesirable condition such as slowing of memory [1]

   b. The field in *Segment Selector* specifies whether the *Segment Descriptor* is included in the *GDT* or in the *LDT* [1]

2. Define following terminologies

   a. reentrant code b. double buffering c. inverted *hash table* *[3]*

3. *Draw the block diagram of cascade mode of 8259 controller* [1]

4. Write the RAID level corresponding to each of the following: [2]

   a) Block level distributed parity : RAID ___

   b) Mirrored: RAID ___

   c) Non redundant: RAID ___

   d) Hamming Code: RAID ___

5. Consider a system with 2 level paging scheme in which a regular memory access takes 200ns and servicing a page fault takes 10ms. An average instruction takes 100ns of CPU time and 2 memory access. The TLB hit ratio is 95% and the page fault rate is one in every 10,000 instructions. What is the effective average instruction execution time? [2]

6. Consider I/O based polling and memory mapped I/O. In what situation would you favour one technique over the other? [1.5]

7. Consider four frames with following page references

   A,B,A,C,D,E,F,A,B,C,F,A,E,F,B,D,E,C,A,F

   Show the content of the frames at every access for the following page replacement policies and Find out the number of page faults. [2]

a. Belady's optimal algorithm b. LRU

8. Suppose you have a 47-bit virtual address space with a page size of 16 KB and that page table entry takes 8 bytes. How many levels of page tables would be required to map the virtual address space if every page table is required to fit into a single page? Be explicit in your explanation and show how a virtual address is structured. [2]

9. The following table represents three different possible designs for memory.

Characteristic	Design1	Design2	Design 3
Physical Memory Address Width	8bit	16bit	32bit
Logical Address Width	12bit	20bit	24bot
Page size	16bytes	32bytes	64bytes

a) For each design, find the maximum number of pages each process can have in logical address space. [1]

b) Which design is likely to have the highest internal fragmentation (if any)? [1]

c) Which design is likely to have the highest external fragmentation (if any)? [1]

10. Briefly describe the steps taken to read a block of data from the disk to the memory using DMA controlled I/O. [2]

11. Provide the detailed steps involved in performing an I/O operation using interrupt mechanism in an x86 architecture [2]

12. Suppose that we have a 64-bit virtual address uses segmented paging, which split as follows:

6Bits	11Bits	11Bits	11Bits	11Bits	14Bits
[Segment ID]	[Level1]	[Level2]	[Level3]	[Level4]	[Offset]

a. How many segments are in this system? [0.5]

b. Assume that the page tables are divided into page-sized chunks (so that they can be paged to disk). How much space have we allowed for a PTE in this system? [0.75]

c. What is the page table overhead for a process that uses one segment? Explain. [1.25]

Department of Computer Science & Engineering

Roll No...B130055C

NATIONAL INSTITUTE OF TECHNOLOGY CALICUT
Monsoon Semester 2015
End Semester Examination
CS 3003 OPERATING SYSTEMS

Duration: 1hour30mnts **PART A** **Maximum Marks: [25]**

1. Suppose a system uses priority scheduling (under the following process load), where a small integer means a high priority [2.5]

Pi	Service time(Pi)	Priority
P0	80	3
P1	20	1
P2	10	4
P3	20	5
P4	50	2

Compute the following

   a. Create a Gantt chart illustrating the execution of these processes
   b. What is the turnaround time for process P2 under priority scheduling?
   c. What is the average wait time for the processes?

2. How many numbers of processes will be spawned after executing this program? *Justify your answer* [1]

```
#include <stdio.h>
#include <unistd.h>
int main()
{       fork();
        fork() && fork() || fork();
        fork();
        printf("forked\n");          return 0;    }
```

3. What is context switching? Suggest an idea to reduce the context switching time? [2]
4. Draw the process state diagram [1]
5. Suppose we want to synchronize two concurrent processes P and Q using binary semaphores S and T The code for process P and Q is shown below:Synchronization statements can be inserted only at points W,X,Y,Z. Which of the following will always lead to an output starting with 001100110011? [1]

PROCESS P	PROCESS Q

While(1){
W:
print 0;
print 0;
X:
}
	While(1){
	Y:
	print 1;
	print 1;
	Z:
	}

a) P(S) at W,V(S) at X ,P(T) at Y,V(T) at Z, S=T=1 initially

b) P(S) at W,V(T) at X ,P(T) at Y,V(S) at Z, S=1 and T=0 initially

c) P(S) at W,V(T) at X ,P(T) at Y,V(S) at Z, S=0 and T=1 initially

d) P(S) at W,V(S) at X ,P(T) at Y,V(T) at Z, S=T=0 initially

6. Compare a counting semaphore and a binary semaphore [1]

7. A semaphore s is a nonnegative integer variable changed or tested only by one of two indivisible access routines V(s) & P(s). Define V(s) and P(s) functions for a data type semaphore S; [2]

8. The test-and-set instruction (TS) is an instruction used to write to a memory location and return its old value as a single atomic. Implement a general semaphore using this test and set instruction! (P & V functions of semaphore) [2]

9. What are the necessary conditions for a deadlock to happen? [1]

10. Consider a disk in which blocks 2, 3, 4, 6, 8, 9, 10, 11, 12, 15, 17, 19, 25, 26, and 28 are free and the rest of the blocks are allocated. Prepare a bit vector (for free space management) for the above scenario? [1]

[0.5+0.5]

11. Answer the following

a. What is i-node(index node) in Unix

b. What will be content of a i-node, corresponding to a device? Brief them

12. Suppose the disk queue contains a set of references for blocks on tracks 76,124,17,269,201,29,137 and 12. Assume disk head is currently pointing to cylinder 57. Perform the SCAN Disk optimization and find how many tracks the head will move across as it serves the request [2]

13. Banker's algorithm is the best known of the deadlock avoidance strategies. Given following details such as alloc table, Resource type C, maximum claim table. Find given allocation is safe or unsafe with the help of banker algorithm. If safe provide the order of execution also [3]

	MaxC ABCD	Allocation ABCD	Available Resource Type – C
P0	3211	2010	A B C D
P1	1202	1100	6 4 4 2
P2	1120	1100	
P3	3210	1010	
P4	2101	0101	

14. What is the importance of VFS switch in Linux? [1]

15. Mention 3 different file allocation techniques [1]

16. Explain the algorithm used for *Open( )* system call with the help of the necessary datastructure [1.5]

17. Draw the sequence of client server establishment using socket related system call [1]

Roll No : B120329CS Name ADITYA KUMAR

NATIONAL INSTITUTE OF TECHNOLOGY CALICUT
Second Midterm Examination Monsoon Semester 2014

Max Marks : 10 **CS3003 – OPERATING SYSTEMS** **(Part - B)**

6½

Answer in the space provided

**** For Questions 1-3 identify if the statements given are TRUE or FALSE. Marks will be awarded only if proper justification is given.***

1. In a demand paging system, if a virtual page number P generates a TLB miss, then the corresponding frame number for P is guaranteed to be found in the Page Table Entry. (1 mark)

TRUE → Reason → Page Table of a process contains all
FALSE → where may be a case of page fault when we have to brough a frame from disk to main m/y & then update the entry in PT.

2. The accuracy of working set depends on the window size. (1 mark)

FALSE |WS| = min (Δ, N) → no. of elements (distinct) elements in a set.

3. Adding prepaging to page-fetch algorithms can increase CPU utilization. (1 mark)

FALSE → Pre-paging reduces the page fault rate for a particular process. To increase CPU utilization, we must increase the degree of multi-programming

4. Considering a page size of 1 KB and that each page table entry takes 4 bytes, how many levels of page tables will be required to map a 34-bit address if every page table fits into a single page. Clearly justify your answer. (2 marks)

Page size = 1KB
no. of frames (inner) = $2^{34}/1KB = 2^{24} = 2^4 M = 16M$
Page table size = 16 × 4 = 64 M → more than size of page
Top-level(1) = $\frac{64M}{1KB} = \frac{2^6 \cdot 2^{20}}{2^{10}} = 64K$ → more (entries)
Size = 64 × 4 = 256 K > more
Top level(2) = 256K/1K = 256 bytes (entries)
Size = 256 × 4 = 1024 B = 1K → fits
∴ 3 levels

5. In a demand paged memory, 200 nanoseconds are required to satisfy a memory request if page is in memory. It takes 8 milliseconds to service a page fault if an empty frame is available or if the page to be replaced is not modified; and 20 milliseconds if the page to be replaced is modified. 60% of the time a page fault occurs, the page to be replaced is dirty. What is the effective access time if the page fault rate is 10%? (2 marks)

$T_{mem} = 200nS$ $T_{pf_2} = 20mS$ [Assumed → Page table access in MM + 1 for Frame]
$T_{pf_1} = 8mS$

$T_{effective} = (1-p)[2 \cdot T_{mem}] + p\,[60\%(T_{pf_2} + 2T_{mem}) + 40\%(T_{f_1} + 2T_{mem})]$

$= (1-0.1)[2 \cdot 200nS] + [0.1(0.6(20mS + 2 \cdot 200nS) + 0.4(8mS + 2 \cdot 200nS)]$

$= 360nS + 0.1(0.6 \times 0.200004 + 0.4 \times 8.000400 \times 10^{-3})$

$= 360nS + 0.1(0.1200024 + 3.200160 \times 10^{-3})$

$= 0.012320616$ Sec

$= 12.3$

$T_S = 0.9 \times 4$ hits $+ 0.1[$

$= 3.6 \times 10^{-7} + 0.1(0.012000024 + 3.20016 \times 10^{-3})$

$= 3.6 \times 10^{-7} + 1.52 \times 10^{-3} = 1.5204 \times 10^{-3}$ s

6. Consider the following piece of code whose function is to initialize each element of a 512-by-512 integer array A to 0. (The array is stored in Row-major order):

```
Initialize_1()
{ for (int i = 0; i < 512; i++)
    for (int j = 0; j < 512; j++)
      A[i][j] = 0;
}
```

Assume that the code for this function fits in one page, and the stack also fits in one page. Let the size of an integer be 4 bytes. Also, the page size is 4096 and the TLB has 4 entries. (Assume LRU replacement policy)

i) Calculate the number of TLB misses. (1.5 marks)

code – 1 page

stack data – 1 page

data = $4 \times 512 \times 512 = 2$ MB = 2 pages

PF at every 4th row change

total = $512 \times 512 \times 4 = 1048576$ bytes = 256 pages

entries of 2 rows can be present at a time.

Therefore, no. of TLB misses =

1 row data = $512 \times 4 = 2048$

∴ at a time 2 rows data can be in 1 page. Since 2 pages are vacant for data.

TLB misses = $512/4 = 128$

ii) Calculate the number of TLB misses if the code is as given below, considering all the assumptions above. (1.5 marks)

```
initialize_2()
{ for (int i = 0; i < 512; i++)
    for (int j = 0; j < 512; j++)
      A[j][i] = 0;
}
```

Page fault occur more frequently now. Due to A[j][i]

↳ this takes data from every other row and causes page fault in every 4th execution.

∴ Total MISSES = $(512 \times 512)/4$

$= 65536$

Roll No: B120329CS Name ADITYA KUMAR

National Institute of Technology Calicut

Department of Computer Science & Engineering

CS3003 – OPERATING SYSTEM

Second MID Term Exam (Monsoon Semester 2014)

Max. Marks: 10 PART A Time: 1 Hour

1. a) Test and Set instruction reads the contents of the given memory word ***lock*** into the register RX and then stores a nonzero value at the memory address lock (**TS RX, LOCK**). How can this instruction be used to prevent two processes from simultaneously entering their critical regions. Write an assembly language subroutine (X86 based) to implement the P( ) function of a semaphore [1 mark]

Test and set is used under a while condition like {while (Test and set (m)). If 'm' was false initially then m is set to true and next instr is allowed to execute. Meanwhile the other process will go in empty loop onlt and after that only they can execute their operation.

assembly code ←
```
mov eax, S
DEC eax
mov S, [eax]
```

b) An alternative instruction to TS is XCHG, which exchanges the contents of two locations atomically. Build an assembly language subroutine to implement the P( ) utilizing XCHG instruction (XCHG RX, LOCK) [1 mark]

```
mov eax , RX
mov ebx , LOCK
mov ecx , RX
mov RX , [ebx]
mov LOCK , [ecx]
```

2. Write 2 examples of socket communication applications [1 mark]

1. Distributed systems
2. Server-client system

3. Shared memory IPC shares the memory in user space! [0.5 mark]
4. Specify two important advantage of thread over process? [1 mark]

① Process creation is costlier than thread creation.
② Context switch in case of thread is faster.

X

5. Write pseudo code to simulate & solve following problems using semaphore

a) A bridge which allows traffic in both directions has a maximum capacity of 3 vehicles at a time. There are vehicles from Both sides(SideA, SideB) competing to pass through the bridge. [1 mark]

```
Semaphore max = 3;
Side vehicle SideA{
    P(max)
    : // pass
    V(max)
}
Side B {
    P(max)
    : // pass
    V(max)
}
```

b) Perform data communication through a half duplex channel of capacity 5(packets) . The channel is used by two processes. Both the processes(process1 and process2) are only sending the packets through the channel. Make sure that the following conditions are met during the data communication:

1) The process1 sends data in one direction and the process2 sends data in another direction of the channel.

2) If the channel is full or it is not the process's direction of communication, the process waits. [1.5 marks]

Half-duplex → Sending in one direction

```
#include <stdio.h>
#include <sys/types.h>
#include <stats.h>
int main(){
    int id1 = mkfifo("Process1", 0);
    id1 = Open("Process1", O_WRONLY);
    int id2 = mkfifo("Process2", 0);
    id2 = Open("Process2", O_RDONLY);
    buf packet[5]; buf receive[5];
int a = write(id1, &packet, 5);
    close(id1);
a = read(id2, &packet
    close(id
```

```
struct buf {
    int x
};
```

```
a = read(id2, &receive, 5);
```

DONE at LAST. PAGE

6. Which signal will be sent to kill a process forcefully? [0.5mark]

SIGKILL

7. Write three ways to register and handle a signal in Linux [0.5 mark]

signal (signal_int , SIG_IGD)
signal (signal_int , SIG_DEF)
signal (signal_int , Signal_handler)

8. Write 2 different ways to delete a message queue whose id is 65536 [0.5 mark]

① msgctl (65536, IPC_RMID, 0);
② msgdt (65536);

9. Write a C program to catch the hang up signal send by the terminal controller, and then print the message "IT IS A HIGH PRIORITY JOB" for once. Application should hang up from second time onwards when you close the terminal. [1.5 Marks]

```
#include <stdio.h>
#include <signal.h>
void * signal_hand(int signumber){
    printf("IT IS A HIGH PRIORITY JOB");
    sleep(2);
    signal(SIGHUP, SIG_DEF);
}
int main(){
int main(int arg, int args[3]){
    signal(SIGHUP, signal_handler);
    return 0;
}
while(1)
{
sleep
}
```

Roll No: B120329CS Name ... Aditya Kumar

National Institute of Technology Calicut

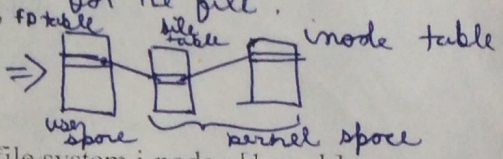
Department of Computer Science & Engineering

CS3003 – OPERATING SYSTEM

First MID Term Exam (Monsoon Semester 2014)

Max. Marks: 11 PART A Time: 1 Hour

1. a) For user mode to kernel mode switching, Linux OS uses System calls [0.5 +.05 Mark]
   b) How does the parameter from user area pass to kernel area while executing the system call using registers eax, ebx, ecx, edx, esi, edi
2. Open ( ) system call return an integer number in case of a successful file opening.
   a) What does this number indicate? [0.5 + 0.5 mark]
   file descriptor index in file descriptor table for the file.
   b) How does kernel map this number to actual data block?
   using i-node table entry ⇒



3. Which are the extra fields that an incore-inode keep than a file system i-node [1 mark]
   Status – locked / unlocked
   updated or not
   i-node number
   reference counts
4. namei algorithm converts the file name to its corresponding i-node number to locate the data block in the file system. namei [0.5 mark]
5. Register context of a process is comprised of values in the registers namely SP, IP, eax, etc. [1 mark]
6. Write a C program to implement a cp command using system calls [2 marks]

```
# include < stdio. h>
# include < unistd. h>

int main ( ) {

    char *from = "Inputfile"; // assumed to be source of copy
    char *to   = "output";    // output file

int id1 = open ( from , O_READ);
int id2 = open ( to , O_WRITE);
  char buffer[1];   Char c1, c2;
while ( id1 ≠ EOF) {
                    read ( id1, &c1, 1);
                    write ( id2, &c2, 1);
                  }

close (id1);
Clsse (id2);
return 0;   }
```

7. Predict the output of the following [2 marks]

a)
```
#include<stdio.h>
#include<stdlib.h>
#include<unistd.h>

int main()
{
    pid_t child;
    child = fork();
    switch(child){
        case -1 :
            perror("fork");
            exit(1);
        case 0 :
            sleep(10);
            printf("%d\n",getppid());
            break;
        default :
            break;
    }
    return 0;
}
```

parentid is printed also child becomes orphan here.

o/p → parent_id

b)
```
#include<stdio.h>
#include<stdlib.h>
#include<unistd.h>

int main()
{
    pid_t child;
    int status;
    child = fork();
    switch(child){
        case -1 ;
            perror("fork");
            exit(1);
        case 0 :
            printf("%d\n",getppid());
            break;
        default :
            printf("%d\n",getpid());
            wait(&status);
            break;
    }
    return 0;
}
```

parentid is printed twice on two separate lines.

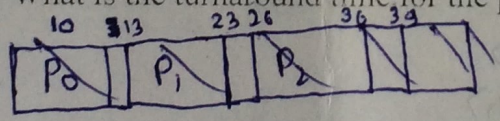
o/p → parent_id
parent_id

8. Assume you have following jobs to execute with one processor. Suppose system uses RR Scheduling with time slice as 10 units. Assume context switch time is 3 units.

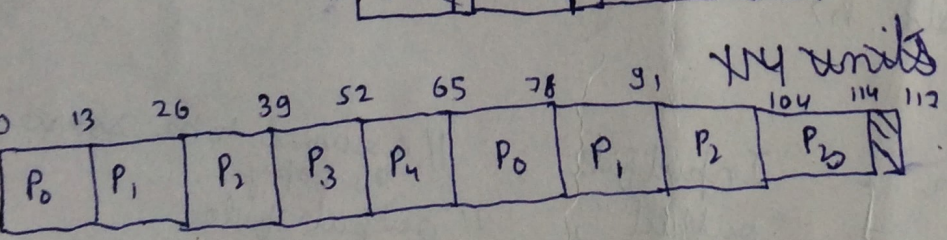
Slice = 10 units
context = 3

Pi	Service time(Pi)
0	75
1	40
2	25
3	20
4	45

a) What is the turnaround time for the process P3? [1 mark]



114 units



114 units

b) What is the ~~average~~ wait time for P3? [1 Mark]

39 units

c) What is the effect of increasing the time quantum to an arbitrarily large number for Round Robin scheduling? [0.5 Mark]

Both, avg. turnaround time and avg. wait time would increase.

Roll No : B120329CS Name : Aditya Kumar

NATIONAL INSTITUTE OF TECHNOLOGY CALICUT
First Midterm Examination Monsoon Semester 2014
CS3003 – OPERATING SYSTEMS
PART - B

Max Marks : 9

Answer in the space provided

7½

1. Consider a system with 32-bit logical addresses, 16-bit physical addresses, and 1KB page size. Compute the Page Table size in bytes if each page table entry takes 4-bytes. Justify your answer (1 mark)

1KB = 2^{10} bytes, if each entry takes 4 bytes then, total no. of entries in each page = $\frac{2^{10}}{2^2} = 2^8$ ∴ 8-bits are needed for offset
we are left with 32 − 8 bits = 24 bit
∴ size in bytes = $2^{24} \times 1 \times 2^{10} = 2^{34}$ bytes

2. Consider a segmentation-based system. At some point in the system operation, the main memory has holes (ie, free segments) in this order : 20KB, 15KB, 90KB, 54KB, 10KB, 30KB, 56KB. A new request for memory of size 22KB arrives. Which hole will be allocated in each of the schemes? Justify your answer. (1 mark)

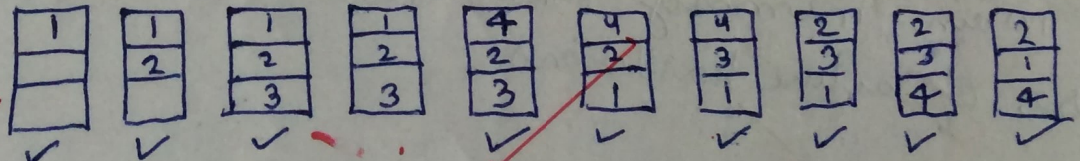
A) Best-Fit : 30KB hole.
Reason :- Internal fragmentation is least in this case. Su is just enough to be hold in 30KB hole.

B) First-Fit : 90KB
Reason :- It is the first free block which can accomodat 22KB

3. Which type of fragmentation(s) does the following memory management schemes suffer from? (2 marks)

A) Paging - Internal fragmentation
B) Segmentation - External fragmentation
C) Buddy systems - Internal fragmentation as well as external.
D) Fixed Partitioning- Internal and external fragmentation.

4. Consider a virtual page reference string: 1, 2, 3, 2, 4, 1, 3, 2, 4, 1 for a demand paging system running on a computer with main memory size of 3 page frames, which are initially empty. Draw the frame allocation-replacement diagrams for LRU page replacement policy and identify the number of page faults. (2 marks)



no. of page faults = 9.

5. Consider a system with 128MB of physical memory, of which the first 8MB is allocated to kernel. The remaining 120MB can be allocated to processes in contiguous chunks (dynamic partitioning). The creation times, finishing times, and memory requirements of 5 processes are given below:

PROCESS	START TIME	FINISH TIME	MEMORY (in MB)
P1	10:01AM	10:05AM	70
P2	10:02AM	10:10AM	30
P3	10:04AM	10:14AM	30
P4	10:06AM	10:15AM	10
P5	10:12AM	10:25AM	100

120 MB m/y

Assume the requests are granted in FIFO order according to First-fit algorithm and each process occupies memory allocated to it until it terminates.

i) Write down the exact time when memory will be allocated to each process. (2 marks)

P1 is allocated memory at ....10:01....AM 10:01

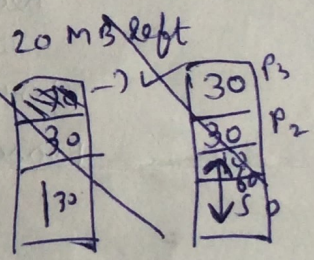
P2 is allocated memory at ....10:02....AM 10:02

P3 is allocated memory at ....10:05....AM 10:05

P4 is allocated memory at ....10:06....AM 10:06

P5 is allocated memory at ....10:14....AM 10:15

70 30



ii) Indicate the times when external fragmentation is observed in the system. Justify. (1 mark)

External fragmentation is never observed

(1) at 10:04

No external fragmentation observed. Every process that wait is due to space limitation of the main memory and not due to the presense of active partitions.

TANVI SHARMA
B090826CS
B Batch

27a

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING
I midterm examination winter 2011-12
CSU313-Operating systems

A

Time: 1 Hour **Max. Marks: 20**

1. The modern systems prefer interrupt driven I/O for slow I/O devices? Why this approach is beneficial for operating systems? (2)

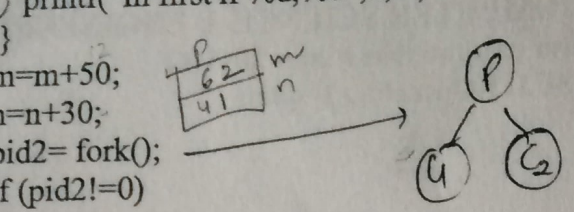
Ans: The above defined statement is carried out to maximise the CPU utilisation time. Without interrupt driven I/O; suppose CPU is executing a process; and it request for an I/O device; then the CPU will wait till the request is executed & being idle for this period (in terms of sec as slow I/O devices); the CPU utilisation is very low. Hence wastage. But in interrupt driven I/O; when it request for an I/O device; the CPU switches to the next process to utilise that time. Once the I/O request is over; interrupt occurs at CPU; CPU leaves the process it is executing (saves) and execute again process 1.

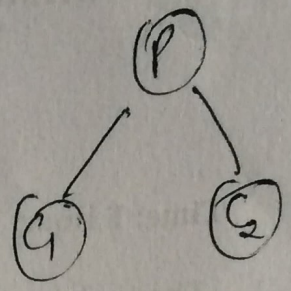
2. What information about a process needs to be saved, changed or updated when a context switch occurs? (3)

Ans: When context switch occurs (saving the current process details and moving to the new process); the PC value of the process; i.e the next command to be executed; accounting information (how much CPU utilisation has already occured); register, process id of the process; process state (running/waiting, etc); are saved of the last process. The registers are loaded for the new process PCB. ~~The PC and IP values due to the~~ In case of an interrupt; the PC and IP values are updated in the interrupt driven vector table for the corresponding interrupt vector. Now the program counter changes for the new process to be executed.

3.
```
int main()
{
 int n,m;
 pid_t pid1,pid2,pid3;
 n=11;
 m=12;
 pid1=fork()
 if(pid1==0) child
 {
  execv("edittext.exe",NULL);
  n=n+15;
  m=m+10;
```

```
① printf("In first if %d,%d",n,m)
}
m=m+50;
n=n+30;
pid2= fork();
if (pid2!=0)
{
   m=m+5;
   n=n+5;
   pid2=wait(NULL);
② printf("Second printf %d,%d",n,m);
}
③ printf(" Last printf %d,%d",n,m)
   exit(0)
}
```





In the above program

a) How many times the first printf() statement is executed and what are the values of n, m ? Justify.

Ans: First printf() statement is not executed even once. Executed zero times. As the execv() command for the child implies that child will execute some other code which is edittext.txt; Hence will not even execute the following subcode :→ n=n+15;
m=m+10;
printf("In first if %d,%d", n, m)
So, No values of m, n will be printed. Values actually are still the declared ones.

b) How many times the second printf() statement is executed and what are the values of n, m ? Justify.

Ans: It will be executed only once by parent as the first child is already executing some other code; and only left is parent for which pid2 ≠ 0 (given condition); The printed values are:→ Second printf 46 67 (by parent)

But the parent will wait for any of the two children to finish execution due to pid2 = wait(NULL); before printing.

c) How many times the last printf() statement is executed and what are the values of n, m ? Justify.

Ans: Two times; one time by parent and one time by the second child.

then parent execute :→ Last printf 46 67

then child2 execute :→ Last printf 41 62

for parent; first after the second printf statement, third printf statement will be executed and hence values will be same.

But the third printf statement is the first time the 2nd child(C_2) is executing any command and as just before forking C_2; the m=62, n=41; so those values will be printed.