Name of the Examination: 6th Semester CST Mid-Term Examinations, 2021

Name of the Subject: Operating System. Subject Code: CS 601

Date of Examination: 20-04-2021

Name of the Student _____

Examination Roll Number_____

G Suite ID_____ Number of sheets uploaded _____

Full Marks: 30 **Time: 45 min.**

- **Answer a maximum of 7 questions**. The full marks for the first 5 question is 30 while the rest of the questions gets you 22 only. So, the first 5 questions are quasi-mandatory.
- Both machine-printed and hand-printed answer scripts will be accepted.
- For figures, if any, draw it (No copy from any source) and import on your answer script
- **YOUR SIGNATURE MUST BE IMPORTED or handprinted** at the end of the script

---

1. Five batch jobs. A through E, arrive at a computer centre at almost the same time.

JOB Name	Estimated run time
A	10
B	6
C	2
D	4
E	8

Find out the average turn-around time for (a) Round robin (time quanta 2 unit) (b) Priority scheduling (priority; 3, 5(H), 2, 1 (L), and 4 for A to E, respectively); (c) First-come, first-served (run in order 10, 6, 2, 4, 8). (d) Shortest job first. For (b) through (d), assume that only one job at a time runs, until it finishes. All jobs are completely CPU bound. [4]

==========

A	B	C*	D	E	A	B	D*	E	A	B*	E	A	E*	A*
0	2		6				…	16			22			28

30

a) RR Tavg = (30+22+6+16+28)/5 = 102/5 = 20.4

D	B	E	A	C
0	6	14	24	26

30

b) PR Tavg = (24+6+26+30+14)/5 = 100/5 = 20

A	B	C	D	E
0	10	16	18	22

30

c) FCFS Tavg = (10+16+18+22+30)/5 = 96/5= 19.2

C	D	B	E	A
0	2	6	12	20 30

d) SJF Tavg =(2+6+12+20+30)/5 = 70/5 = 14

2. Using *syscall* function Linux (X86-64) can write (1) or _exit (60) as shown in the following assembly language program segment – write the corresponding C program. [5]

```
;-------------------------------
.section .data
String:
.ascii “hello, world\n”
String_end:
.equ    len, string_end – string
.section        .text
,globl  main
        Movq    $1, %rax
        Movq    $1, %rdi        ; stdout has descriptor 1
        Movq    $string, %rsi
        Movq    $len, %rdx
        Syscall
        Movq    $60, %rax
        Movq    $0, %rdi
        Syscall
==============================
```

It is clear that the ascii string is being written [func. No 1. in rax, output device (default is monitor) in rdi, rsi holds string pointer and rdx holds the string length. And it is making an exit through _exit call; so the corresponding C program will be

```
main() { printf(“hello, world\n”); _exit(0);}
```

3. Consider the following program segment (assume that appropriate header files have been included) and find out the number of output lines for N = 28; provide a C program segment to find out the number of lines of output. [6]

```
int main() { int i;

  for(i=0; i<N; i++){
    fork();
    printf("Hello World\n");}
  return 0;}
```

---

every fork() doubles the count – the corresponding program in C is

```
main() { int I, s=0, t=2; for (i=0; i<28;i++){s += t; t*=2;} return 1;}
```

No. of output lines would be the sum of the GP series a =2, n = 28, r = 2 where

S = a(r^n -1)/(r-1) = 2(2^28 – 1)/(2-1) = 2^29 – 2 = 536870910

4. The contents of IA-32 logical (Linear 32-bit) address 3013D5H is 5651E8D2H (in little endian form). Assume that; i) CR3 (Page directory base pointer) = 0344C000H ii) [0344C000] = 1CA6B867H

Find the address of the PDE (Page Directory entry) and the address of PTE (Page table entry). If the PA = 19AC75D5H then find out the contents of the PTE and the contents of the physical address i.e., [PA].
[Note each PTE entry is 4 bytes long]
[8]

In IA-32 the 32-bit LA is divided into 3-parts 10bit offset for the page directory table, 10 bit for the page table and 12-bit offset for the page-frame. In our case we have the LA in binary

0000 0000 00|11 0000 0001| 0011 1101 0101 (LA = 003013D5H)

[divided into <10bit><10bit><12bit> form]

I) Address of the PDE entry 0344C000 + 0000 = 0344C000h
II) Address of the PTE entry 67B8A61C + 301 X 4 [AS WIDTH OF EACH PTE is 4] = 67B8A220H
III) PA = 19AC75D5H; SO THE CONTENTS OF THE PTE IS 19AC75D5 – 3D5 (OFFSET GIVEN BY LAST 12 BITS IS 3D5) = 19AC7200H
IV) The contents of the Logical address (LA) and PA would be same so we have the value D2E85156H in the physical memory.

5. You are given the following data about a virtual memory system: (a)The TLB can hold 1024 entries and can be accessed in 1 clock cycle (1 nsec). (b) A page table entry can be found in 100 clock cycles or 100 nsec. (c) The average page replacement time is 6 msec. If page references are handled by the TLB 99% of the time, and only 0.01% lead to a page fault, what is the effective average address-translation time for 10000 clock cycles? [7]
   ==========
   Average Att = (TLB translation + Page Table translation + page fault replacement)/3 ns
   = (9900x1+ 99 x 100 +1x6x1000x1000)/3 = 6019800/3 ns = 2.006 (approx.) ms

6. Which function is used to reset the top of the heap in LINUX? [2]
7. In LINUX process address space implemented in 32-bit architecture wherefrom RO code segment and the stack segment start? [2]
8. Draw the process context switch diagram (Process A – made a disk read) process B is taking over and execution control returns to process A after the Disk interrupt on completion of the disk read. Diagram should clearly show the execution flow through the user and kernel code. [2]
9. While running a program in a UNIX system you have got the message “core dumped” … what do you mean by this message. Also, name at least 2 signals whose actions are “terminate and core dump”.
   [3]
10. Elaborate on the reasons for which thread synchronization is necessary [2]
11. Name the basic IPC mechanisms and their characteristics [2]
12. For solution to critical section problem a technique must satisfy three basic criteria. Name and elaborate them. [2]
13. What is a TLB and why is it used in a Paging mechanism?

What are hard and soft TLB misses. [3]

14. Explain internal and external fragmentation. [2]
15. Draw the transition diagram of processes in a system considering ready, run and states. [2]

Signature of the student