Indian Institute of Technology Kharagpur

SPRING Semester, 2016
COMPUTER SCIENCE AND ENGINEERING

CS31702: Computer Architecture and Operating System

Mid–semester Examination

Full Marks: 50

Time allowed: 2 hours

INSTRUCTIONS: **Special credit would be given for answers which are short and to–the–point. This question paper has two pages. ANSWER ALL QUESTIONS.**

1. (a) Study the following recursive C function to calculate and return the value of an argument incremented by 1, and write an equivalent recursive MIPS function. Note that the function can handle both positive and negative integers as argument. You are allowed to use pseudoinstructions. NO CREDIT WILL BE GIVEN FOR A NON-RECURSIVE IMPLEMENTATION. (8 marks)

```
/* The following function returns y+1 */
int increment (int y)
{
   if (y == 0) return 1;
   else if (y % 2 == 1) return (2 * increment(y/2));
   else return (y+1);
}
```

(b) A student wanted to load the constant 0x1234abcd in the register `$t0`. For this, he wrote the following MIPS code:

```
lui  $t0, 0x1234
addi $t0, $t0, 0xabcd # instead of the more common ori $t0, $t0, 0xabcd
```

However, when the student assembled and executed the program, he did not find the expected value in register `$t0`. Can you explain what went wrong? (2 marks)

(c) Write three salient features of the MIPS instruction set. (3 marks)

(d) Why does modern processor design focus more on architectural enhancements in place of increase in clock frequency, to improve processor performance? (2 marks)

(e) Distinguish between RAID4 and RAID5. (2 marks)

(f) A designer wants to improve the overall performance of a given machine with respect to a target benchmark suite and is considering an enhancement X that applies to 50% of the original dynamically-executed instructions, and speeds each of them up by a factor of 3. The designer's manager has some concerns about the complexity and the cost-effectiveness of X and suggests that the designer should consider an alternative enhancement Y. Enhancement Y, if applied only to some (as yet unknown) fraction of the original dynamically-executed instructions, would make them only 75% faster. Determine what percentage of all dynamically-executed instructions should be optimized using enhancement Y in order to achieve the same overall speedup as obtained using enhancement X. (Hint: use *Amdahl's Law.*) (5 marks)

2. (a) What are the situations under which an unfinished process might need to leave the CPU? Explain. (3 marks)

(b) Explain why many modern operating systems such as UNIX and Windows do not have any long-term scheduler. (2 marks)

(c) Explain the concept of *Vectored Interrupt*, and mention its advantage over a single monolithic *Interrupt-handler Routine* to handle all possible interrupts. (4 marks)

(d) A program repeatedly performs a three-step process: it reads in a 4 kB block of data from disk, does some processing on that data, and then writes out the result as another 4 kB block elsewhere on the disk. Each block is contiguous and randomly located on a single track on the disk. The disk drive rotates at 7200 RPM, has an average seek time of 8 ms, and has a transfer rate of 20 MBps (megabytes per second). The controller overhead is 2 ms. No other program is using the disk or processor, and there is no overlapping of disk operation with processing. The processing step takes 2×10^7 clock cycles, and the clock rate is 400 MHz. What is the overall speed of the system in blocks processed per second, assuming no other overhead? (5 marks)

3. (a) Explain the problem of *starvation* associated with *priority scheduling*, and one possible solution to the problem. (3 marks)

(b) *Preemptive SJF scheduling*, sometimes called *shortest-remaining-time-first scheduling* is a relatively aggressive scheduling policy, whereby the currently executing process might be preempted, if the new CPU burst of a newly arrived process is shorter than whet is left of the current CPU burst of the current process. Suppose four processes P_1, P_2, P_3 and P_4 arrive at the ready queue at times 0, 1, 2 and 3 respectively, with burst durations 8, 4, 9 and 5 respectively, in a system that uses preemptive SJF scheduling. Draw (i) the resulting Gantt chart, and (ii) calculate the average waiting time for the processes. (6 marks)

(c) Consider a variation of the RR scheduling scheme where the entries in of the ready queue are pointers to the PCBs. (i) What would be the effect of having multiple pointers pointing to the same PCB? (ii) How can you achieve the same effect in classical RR scheduling? (iii) Does this modified scheme remain RR scheduling any more in essence? Explain. (5 marks)