**MT260 Computer Architecture and Operating Systems**

**Specimen Exam 2003 Solution**

**Question 1**

(a) passwd

(b) (i) /dev
(ii) /etc
(iii) /bin
(iv) /lib

**Question 2**

(a) \(171.625_{10} = 10101011.101_2\). The IEEE 754 format requires the number to be normalized as \(1.01010111_2 \times 2^7\). Using 8 bit excess-127 notation for the exponent, and not storing the first digit of the mantissa gives the result.

\[
\begin{align*}
0 & 1 0 0 0 0 1 1 0 \\
0 & 1 0 1 0 1 1 1 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
\end{align*}
\]

(b) (i) \(64_{10} = 0 1 0 0 0 0 0 2\)
\(13_{10} = 0 0 0 0 1 1 0 1_2\)
\(-13_{10} = 1 1 1 1 0 0 1 0_2 + 12 = 1111 0011_2\)

\(64_{10} - 13_{10} = 0100 0000_2 + 1111 0011_2\)

carry out: \(1 0011 0011_2\)

(ii) \(41_{16} = 0100 0001_2\)
\(-41_{16} = 1011 1110_2 + 12 = 1011 1111_2\)

\(40_{16} = 0100 0000_2\)
\(-40_{16} = 1011 1111_2 + 12 = 1100 0000_2\)

\(-41_{16} - 40_{16} = 1011 1111_2 + 1100 0000_2\)

carry out: \(1 0111 1111_2\)

(iii) \(7_{10} = 0 0 0 0 0 0 1 1 1_2\)
\(-7_{10} = 1 1 1 1 1 0 0 0 2 + 12 = 1111 1001_2\)
\(8_{10} = 0 0 0 0 0 0 1 0 0 2\)

\(-7_{10} + 8_{10} = 1111 1001_2 + 0000 1000_2\)

carry out: \(1 0000 0001_2\)
**Question 3**

(a)  
(i) -127  
(ii) -128  

(b) Assume $x$ is the hit rate,  
$$20x + 100(1-x) = 60$$  
$$20x + 100 - 100x = 60$$  
$$40 = 80x$$  
$$x = \frac{40}{80} = 0.5 \text{ or } 50\%$$  
[Either answer $40/80$, $50\%$, $0.5$ is acceptable]  

(c) (1) Branch processing unit; (2) integer processing unit; (3) load/store unit; and (4) floating point processing unit. [3 out of 4]  

(d) Input 2 numbers and output their positive difference.  
In other words, if $A$ and $B$ are the two integers, output $|A-B|$.

**Question 4**

(a)  
```
wait(entry);
if count = n
    then exit;
count := count + 1;
if count > 1
    then
        begin
            signal(entry)
            wait(wait);
        end
else
    signal(entry);
```

```
signal(barber);
    “shave”
wait(entry);
count := count – 1;
if count > 0
    then
        signal(wait);
signal(entry);
```
(b) (i) \[ 80 \rightarrow 74 \rightarrow 68 \rightarrow 17 \rightarrow 0 \rightarrow 95 \rightarrow 100 \rightarrow 128 \]

head movement = \( \frac{80 + 128}{2} = 208 \)

you may stop here

(ii) \[ 80 \rightarrow 74 \rightarrow 68 \rightarrow 95 \rightarrow 100 \rightarrow 128 \rightarrow 17 \]

head movement = \( (80-68) + (128-68) + (128-17) = 183 \)

you may stop here

(c) Demand interpret as the maximum needs version:
The current state is unsafe since only one resource unit is available, and at least two units are needed to enable P3 or P7 (eventually all processes) to complete.

Demand interpret as the current needs version:
The current state is unsafe since only one resource unit is available, and at least two units are needed to enable P7 to complete and it is not possible to have a safe sequence.

Either version of the answer is acceptable.

Question 5

(a) To support segmentation, we need a segment table that basically consists of base register - limit register pairs, one for each segment in a process. In translating the address, \( s \) is used to index into the segment table. If \( d < \) value in the limit register then it is added to the value in the base register to form a physical address.

(b) (i) 180
(ii) 1430
(ii) Illegal reference

(c) A device independent system is one in which the physical characteristics of the devices are hidden by means of a generic interface. This can be achieved by means of using a logical device to model a generic device, which is then mapped to different physical devices as needed. The user only deals with the logical device and has no need to handle the physical characteristics of the actual devices.

(d) An interrupt vector is the pointer to an address in memory that contains the program routine for processing the interrupt.

(e) If the gray scale requires a 3×3 matrix, a 1200 dot/inch printer is effectively reduced to \textbf{400 pixels per inch}. In actuality, the gray scale is interpolated, so the resolution is somewhat higher.
Question 6

(a) At the same clock speed, a parallel bus will transfer data faster, since all the data bits in a word are transferred at once.

However, parallel buses are limited to short distances, both by the cost of wire, and by interference between the lines that occurs when the distances get too long.

Data can be transferred over longer distances via serial buses, but at a slower rate.

Serial buses also have the advantage that they may be used as an interconnection method to other media that can only pass data one bit at a time, such as telephones. (A new twist on this idea is the use of fiber optic cable to achieve extremely high transfer rates.)

(b) 100 million instructions per second.

(c)  
- Arithmetic operations;
- Logic operations;
- Data movement;
- Control operations.

(d) The average seek time for a hard disk is the time required to locate a particular track on the disk. The exact location of the track is known, and the head will be moved directly to that position.

The CD-ROM uses a spiral track, so the exact position of the track for a particular angle of rotation is not known.

Furthermore, the number of blocks on a track varies from track to track because the CD-ROM is CLV, so the radial location of the desired block is known only approximately.

To find the desired track, the head must find a track and block in the approximate region of the desired block, read the block number, then use that value to move closer to the desired location. Thus, the seek is a searching operation, which takes time.
Question 7

(a)

**FCFS**

<table>
<thead>
<tr>
<th>Time</th>
<th>0</th>
<th>350</th>
<th>475</th>
<th>950</th>
<th>1200</th>
<th>1275</th>
</tr>
</thead>
<tbody>
<tr>
<td>p1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>p2</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>p3</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>p4</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>p5</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**SJF**

<table>
<thead>
<tr>
<th>Time</th>
<th>0</th>
<th>75</th>
<th>200</th>
<th>450</th>
<th>800</th>
<th>1275</th>
</tr>
</thead>
<tbody>
<tr>
<td>p5</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>p2</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>p4</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>p1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>p3</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**RR (quantum = 50)**

<table>
<thead>
<tr>
<th>Time</th>
<th>0</th>
<th>50</th>
<th>100</th>
<th>150</th>
<th>200</th>
<th>250</th>
<th>300</th>
<th>350</th>
<th>400</th>
<th>450</th>
</tr>
</thead>
<tbody>
<tr>
<td>p1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>p2</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>p3</td>
<td>475</td>
<td>525</td>
<td>550</td>
<td>600</td>
<td>650</td>
<td>700</td>
<td>750</td>
<td>800</td>
<td>850</td>
<td>900</td>
</tr>
<tr>
<td>p4</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>p5</td>
<td>1000</td>
<td>1050</td>
<td>1100</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

| Waiting Time | p1 | p3 | p1 | p3 |

<table>
<thead>
<tr>
<th>Process</th>
<th>FCFS</th>
<th>SJF</th>
<th>RR (quantum = 50)</th>
</tr>
</thead>
<tbody>
<tr>
<td>P1</td>
<td>0</td>
<td>450</td>
<td>750</td>
</tr>
<tr>
<td>P2</td>
<td>350</td>
<td>75</td>
<td>425</td>
</tr>
<tr>
<td>P3</td>
<td>475</td>
<td>800</td>
<td>800</td>
</tr>
<tr>
<td>P4</td>
<td>950</td>
<td>200</td>
<td>700</td>
</tr>
<tr>
<td>P5</td>
<td>1200</td>
<td>0</td>
<td>400</td>
</tr>
</tbody>
</table>

(b)

-- a way to connect the I/O interface and the main memory;

-- the I/O module that is capable of DMA must be able to read data from and write data to the main memory;

-- a means to avoid conflict among the CPU and I/O modules in occupying the memory bus at any given instant.
Question 8
(a) (i) Offset = 12 bits -> Size of each page is $2^{12}$ bytes.
Total number of pages $(2^{12})(2^{8}) = 2^{20}$

(ii) Since the level one page field is 12 bits wide, the memory space of $2^{32}$ bytes is partitioned into $2^{12}$ or 4096 chunks, hence each chunk corresponds to $2^{20}$ bytes or 1 Mbytes. Hence the process will need 16 entries in the level one page table and 16 level two page tables. Hence 17 tables are actually needed to handle the process.

(iii) First note that the top level page table contains $2^{12}$ or 4096 entries. When a logical address is first presented to the memory management system, the level 1 page field is used to index into the top level page table. Each entry in this table handles $2^{32}$ bytes/$2^{12} = 2^{20}$ bytes or 1 Mbytes of logical address space.

Hence consider the 32 bit address given = 00102007H. The address yields the value 1 for the level 1 page table field, 2 for the level 2 page table field, and 7 for the offset.

The memory management unit uses the first offset to obtain entry 1 in the level one page table, which corresponds to address 1M to 2M. Now it uses the level 2 page table field value of 2 to index into the second level page table and extract entry 2, which corresponds to relative address 8192 to 12287 within its 1M address space.

This entry contains the page frame number of the page containing the given virtual address. Assuming that this page is in memory, the page frame number is taken from the second level page table and combined with the offset(7) to construct the physical address.

(b) Tag size is 22 bits, and word size is 2 bits.

(c) A client-server network is one in which network access is controlled by one or more server computers.

- Client computers may only access the network to receive services that are provided by the server computers. Server software can communicate with every computer on the network, but client software can only communicate with the server.
- In a peer-to-peer network, the network software on any computer can communicate with the network software on any other computer on the network.

A peer-to-peer network can be used as a client-server system, but the reverse is not true
Question 9

(a) This program prints the quotient of the two input numbers, A and B.
In other words, if \( A/B = Q \), the output is Q.

\[
\begin{align*}
00 & \text{ IN} \quad ; \text{input A} \\
01 & \text{ STO 19} \\
02 & \text{ IN} \quad ; \text{input B} \\
03 & \text{ STO 18} \\
04 & \text{ LDA 19} \\
05 & \text{ SUB 18} \quad ; \text{perform} \ A - B \\
06 & \text{ BRP 10} \quad ; \text{if} \ A > B \\
07 & \text{ LDA 20} \quad ; \text{otherwise stop and output the quotient} \\
08 & \text{ OUT} \\
09 & \text{ BR 15} \\
10 & \text{ STO 19} \quad ; \text{A <- A - B} \\
11 & \text{ LDA 20} \\
12 & \text{ ADD 17} \quad ; \text{increment} \ Q \text{ by} \ 1 \\
13 & \text{ STO 20} \\
14 & \text{ BR 04} \\
15 & \text{ HLT} \\
17 & \text{ DAT 01} \quad ; \text{a constant value for incrementing the quotient by} \ 1 \\
18 & \text{ DAT 00} \quad ; \text{holds the second number}\ (B) \\
19 & \text{ DAT 00} \quad ; \text{holds the first number}\ (A) \\
20 & \text{ DAT 00} \quad ; \text{holds the quotient}
\end{align*}
\]

(b) LIFO

\[
\begin{array}{cccccccccc}
2 & 4 & 1 & 3 & 4 & 5 & 6 & 5 & 5 & 2 & 6 & 3 \\
>2 & 2 & 2 & 2 & 2 & 2 & 2 & 2 & 2 & 2 & \\
>4 & 4 & 4 & 4 & 4 & 4 & 4 & 4 & 4 & 4 & 4 \\
>1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\
>3 & 3 & >5 & >6 & >5 & >5 & 5 & 5 & >6 & >3 & \\
* & * & * & * & * & * & * & * & * & *
\end{array}
\]

The number of page fault is 9.

(c)

<table>
<thead>
<tr>
<th></th>
<th>CISC</th>
<th>RISC</th>
</tr>
</thead>
<tbody>
<tr>
<td>Instruction Set</td>
<td>Many complex instructions. Lengths and formats are variable.</td>
<td>Fewer and less complex instructions. Lengths and formats are fixed.</td>
</tr>
<tr>
<td>Addressing Modes</td>
<td>Many and flexible addressing modes.</td>
<td>Simple and limited addressing modes. Only load and store instructions can access memory. All other instructions use register-to-register addressing.</td>
</tr>
<tr>
<td>Registers</td>
<td>Only a few registers are provided for program use.</td>
<td>Multiple banks of registers are available for program use.</td>
</tr>
<tr>
<td>Code size:</td>
<td>Fewer instructions are required to perform a given programming task. So, the size of program is generally smaller.</td>
<td>More instructions are required to perform an equivalent programming task. So, the size of the program is generally bigger.</td>
</tr>
<tr>
<td>Code quality generated by compiler</td>
<td>It is easier for a compiler to generate machine code from highlevel language; the code quality is not very dependent on the performance of a compiler.</td>
<td>Since instruction set is simple and limited, the code quality is very dependent on the optimization technique of an RISC compiler</td>
</tr>
<tr>
<td>Key technology</td>
<td>Microcode/microprogramming</td>
<td>Instruction pipelining, superscalar processing.</td>
</tr>
</tbody>
</table>
Question 10

(a) SJF is able to minimize the average wait time of the waiting processes by always selecting the shortest job first.

In practice it has some weaknesses:

(1) not preemptive;
(2) in practice it is seldom possible to predict the job time accurately
(3) user can actually manipulate the system by splitting jobs into smaller jobs and gaining priority. This increases the overhead of the system due to the large number of jobs to be handled.

Bad for time-sharing because non-preemptive; long jobs suffer from poor response time.

(b) The main function of DMA is to transfer large quantities of data between memory and I/O devices as it eliminates the need for the CPU to be involved in the transfer. If it is not in use or not functioning right, the CPU will be tied up for long periods of time.

(c) Location transparency: This means that resources can be accessed without knowing their physical location. Users might need to know the name of resources but not their location.

Access transparency: Local resources and remote resources are accessed in the same way. This means that, to the users, it does not matter whether the resources are from local or remote machines, as the way to access them does not differ.

Migration (mobility) transparency: This allows the movement of resources and users’ submitted jobs within the distributed system without affecting how the users operate the system. This implies that users will not notice a resource or a submitted job has migrated to another machines within the distributed system.

Replication transparency: It allows multiple instances of resources to be present throughout the system without the user or application programmers knowing anything about the replicas. Multiple instances of resources are usually distributed evenly throughout the system so that users can always access one. This increases the reliability and performance of the services provided by the system.

Concurrency transparency: This means that multiple users can use the same resource at the same some without external mechanisms, to avoid interference. The users are not even aware that there are other users who access the same resource. An example of this transparency is multiple processes, or users accessing a shared database simultaneously. The distributed system will take some measures to maintain the consistency of the database, but these are transparent to the users or processes.

Failure transparency: This means that when some failures, such as hardware, software, or communication link failures, have occurred within the distributed system, it still maintains its services as usual. Its users are not aware of any failures within the system.

[any 4 of the above]