1. If the following shell script is stored in an executable file named “dirList” and then is executed by issuing command “dirList”, what would happen? [4]

```bash
PATH=$PATH:/bin:/usr/local/bin:$HOME/bin
# input argument: a directory name
cd $1
echo " 
ls
for i in *
do if test -f $i
    then :;
    else dirList $i
fi
done
```

**Ans:** Displays all the files and the directories in the given directory and the subdirectories recursively.

2. A computer has a cache, main memory, and a hard disk used for virtual memory. If a reference word is in cache, 20 ns are required to access it. If it is in the main memory but not in the cache, 60 ns are needed to load it into cache (this includes the time to originally check the cache), and then the reference is started again. If the word is not in the main memory, 12 ms are required to fetch the word from disk, followed by 60 ns to copy it to the cache, and then the reference is started again. The cache hit ratio is 0.9 and the main memory hit ratio is 0.6. What is the average time in ns, required to access a reference word on this system? [7]

**Ans:** There are three cases to consider:

<table>
<thead>
<tr>
<th>Location of referenced word</th>
<th>Probability</th>
<th>Total time for access in ns</th>
</tr>
</thead>
<tbody>
<tr>
<td>In cache</td>
<td>0.9</td>
<td>20</td>
</tr>
<tr>
<td>Not in cache, but in main memory</td>
<td>(0.1)(0.6) = 0.06</td>
<td>60 + 20 = 80</td>
</tr>
<tr>
<td>Not in cache or main memory</td>
<td>(0.1)(0.4) = 0.04</td>
<td>12ms + 60 + 20 = 12,000,080</td>
</tr>
</tbody>
</table>

So the average access time would be:

\[
\text{Avg} = (0.9)(20) + (0.06)(80) + (0.04)(12000080) = 480026 \text{ ns}
\]

3. Describe the differences among short-term, medium-term and long-term scheduling. Which scheduler can control the degree of multiprogramming? [3 + 0.5 = 3.5]
**Ans:** Short term (CPU Scheduler): Selects from jobs in memory, those jobs that are ready to execute and allocates CPU to them.

Medium-term: used especially with time-sharing systems as an intermediate scheduling level. A swapping scheme is implemented to remove partially run programs from the memory and reinstate them later to continue where they left off. Swapping may be necessary to improve process mix or to reduce the degree of multiprogramming.

Long-term (job scheduler): determines which jobs are bought into memory for processing.

The primary distinction between the two schedulers lies in the frequency of execution. The short-term scheduler must select a new process quite often. Long-term is used much less often since it handles placing jobs in system and may wait a while for a job to finish before it admits another one. Long-term scheduler controls the degree of multiprogramming.

4. What are the benefits and detriments of each of the following? Consider both the systems and programmer’s level

(a) Automatic and explicit buffering

(b) Fixed-sized and variable-sized messages

**Ans:**

(a) Automatic buffering provides a queue with indefinite length, thus ensuring the sender will never have to block while waiting to copy a message. There are no specifications how automatic buffering will be provided; one scheme may reserve sufficiently large memory where much of the memory is wasted. Explicit buffering specifies how large the buffer is. In this situation, the sender may be blocked while waiting for available space in the queue. However, it is less likely that memory will be wasted with explicit buffering.

(b) Fixed-size messages - system level implementation is straightforward however programming tasks tends to be complex; Variable-size messages – more complex system level implementation but programming tasks are simpler.

The implications of this are mostly related to buffering issues; with fixed-size messages a buffer with specific size can hold a known number of messages. The number of variable-sized message that can be held by such buffer is unknown. Consider how Windows 2000 handles this situation: with fixed-size messages (anything < 256 bytes), messages are copied from the address space of the sender to the address space of receiving process. Larger messages are (variable-sized messages) use shared memory to pass the messages.

5. What are the three general methods for passing parameters to the operating system?

**Ans:**

(i) Pass the parameters in registers

(ii) Registers pass the starting addresses of blocks of parameters (parameters are stored in block, or table, in memory)

(iii) Parameters can be placed, or pushed, on to the stack by program, and popped off the stack by the operating system.