Massachusetts Institute of Technology
Sloan School of Management

15.564 **Information Technology I** **Spring 2003**

Problem Set 1

The Basics: Hardware, Software and Operating Systems

Due: Tuesday, February 25, 2003. PLEASE SUBMIT A HARDCOPY OF YOUR HOMEWORK DURING CLASS ON THE DUE DATE

PROBLEM 1: CACHING AND SYSTEM PERFORMANCE (20%)

One of the points emphasized in class is that the performance of a computer system depends heavily on the interaction of several system components. Hence it is very difficult to accurately express the performance by using any single number. One useful measure of the performance of a computer system is the number of machine instructions it can execute per second. This is usually measured in MIPS, which stands for Million Instructions Per Second. The exact *MIPS-rating* for a given machine depends on the program that this machine is running. However, for a given program, the MIPS-rating of different computers provides a useful measure for comparing the relative performance of these systems in executing that particular program.

In this problem, we will calculate the impact of caching on the performance of two different (imaginary) computer systems:

System Name	**Lightning P500**	**Thunder T500**
Processor	500 MHZ Pentium	500 MHZ Pentium
Main Memory	64 MB (access time: 20ns)	64 MB (access time: 20 ns)
Cache Memory	None	256 KB (access time: 2ns)

Suppose that there are five stages of instruction execution (we assume that the address of the next instruction is stored in the instruction counter as part of the execution stage):

1. Fetch instruction from memory
2. Decode instruction (figure out what kind of instruction this is)
3. Fetch operands (from registers or memory, depending on the type of instruction)
4. Perform operation (e.g. add, subtract)
5. Write result (to registers or memory, depending on the type of instruction)

Suppose that executing one stage takes one processor clock cycle (one period of the internal processor clock), unless it involves a memory access. Stages that require main memory access require time equal to the main memory access time in order to complete. Since this is usually more than a clock cycle, main memory accesses slow down instruction execution.

Systems with cache memory can satisfy most memory access requests directly from the cache. Since cache memory access time is usually less than a clock cycle, memory accesses that are satisfied from the cache take one clock cycle as well. Memory accesses that are not satisfied from the cache have to access main memory and take more time, as explained above. For a given program, the *cache hit ratio* is the percentage of memory accesses that can be satisfied directly from the cache.

We will compare the total time it takes the two systems to execute a test application:

Our test application consists of 1,000,000 instructions. Approximately 40% of the instructions fetch their operands and write their results in memory. The rest work exclusively using registers. The cache hit ratio is 85% for operand data accesses and 100% for instruction accesses.

Question I: *How much time does it take system P500 to execute an instruction which does not access main memory?*

Question II: *How much time does it take system P500 to execute an instruction which DOES access main memory?*

Question III: *How much time does it take system P500 to execute the test application?*

Question IV: *What is the MIPS-rating of system P500 based on the test application?*

Question V: *How much time does it take system T500 to execute an instruction which does not access main memory?*

Question VI: *How much time does it take system T500 to execute an instruction which DOES access main memory? Since system T500 provides cache memory, there are two different possibilities. Provide an answer for each of the following two cases:*
a) the memory access can be satisfied from the cache
b) the memory access cannot be satisfied from the cache

Question VII: *How much time does it take system T500 to execute the test application? What is the MIPS-rating of system T500, based on the test application? How much faster is system T500 from system P500 in executing the test application?*

Question VIII: *For which of the following two applications do you think that the MIPS-rating provides a more meaningful measure of performance. Explain your answer.*

a) A differential equation solver, which contains a lot of mathematical calculations performed in memory and very little interaction with I/O devices

b) A database application which needs to read and write the hard disk in almost any transaction with the end users

PROBLEM 2: Text Compression (20%)

In this problem you will familiarize yourselves with a very useful compression utility program called WinZip.

Learn how to use WinZip. If WinZip is not already installed in the computer you're using, you can download it from the Internet. Go to

http://www.winzip.com/WinZip/download.html

and follow the instructions listed on that page.

you will find a number of test files attached to this assignment (Problem Set 1). The following is a list of these files:

`one.txt`	A text file
`random.txt`	Another text file
`icse3.doc`	A Microsoft Word document
`database.mdb`	A Microsoft Access database
`telnet.exe`	An executable program

Question I: How many floppy disks would you need in order to save all test files in their original format? What problem would you encounter if you tried to use floppy disks to save all test files in their original format?

Question II: Create a .zip archive containing a compressed version of all above test files. ***Include the .zip archive as part of your homework submission on a floppy disk OR email it to the TAs.***

We define the *compression ratio* to be the percentage of the original file size that has been eliminated by compression. For example, a compression ratio of 64% means that the compressed file has a size equal to 36% (=100%-64%) of the original file size. WinZip displays the compression ratio for each file in an archive under column "Ratio". As you can see by creating the previous archive, the compression ratio can vary widely from file to file.

Question III: Examine test files 1 and 2. Although they are both text files, they have very different compression ratios in the archive that you have constructed. After examining the contents of both files, give a one paragraph explanation of why WinZip was able to compress one file much more efficiently than the other.

PROBLEM 3: Digital Video Representations (10%)

Question I: How many seconds of uncompressed, full-motion video can fit in a CD-ROM? Assume that full-motion video corresponds to a screen resolution of 640x480 with 256 different colors and 30 frames per second. A CD-ROM can fit up to 650MB of data.

Question II: Answer the same question for a DVD. One-sided, double-layered DVDs can fit up to 8.5GB of data.

Question III: What is the minimum compression ratio that is required in order to be able to fit an entire feature-length (120min) movie on a one-sided DVD?

Question IV: A 4x DVD drive has an average data transfer rate of about 5MB/sec. Based on that number and the data storage requirements for full-screen video that you calculated in the previous questions, can you see another important benefit of video compression (other than allowing entire movies to fit on a single disk)?

PROBLEM 4: Disk Fragmentation (20%)

As we mentioned in class, when Windows or UNIX is saving a file, each successive data block is written to the first unoccupied cluster found on the disk. Initially disks are empty and thus successive data blocks are written to physically adjacent disk clusters. After several iterations of file deletion and creation, however, there might no longer be any block of adjacent free clusters big enough to fit an entire file left on our disk. Files then become *fragmented*, that is, parts of a file are physically scattered in various different places on the disk. A File Allocation Table (FAT) is used to keep track of where each successive file data block is physically stored (see lecture slides). Nevertheless, file fragmentation can significantly increase the time required to read a file and thus reduce the overall system performance. For that reason, periodically, it makes sense to use a special utility program to *defragment* the disk, that is, rearrange data on the disk so that files are stored on contiguous clusters. Windows includes such a program in its standard distribution.

In this problem you will get a sense of the performance reduction that can be caused by file fragmentation. Consider a hard disk with the following parameters:

Total capacity:	10 Gbytes	amount of data that can be stored in the entire disk
Capacity per track:	512 Kbytes	amount of data that can be stored in a single track
Average seek time:	12 ms	average time to move head from one track to a different (non-adjacent) track
Minimum seek time:	3 ms	time to move head from one track to the immediately adjacent track
Average latency:	6 ms	average interval elapsing from the time the disk head arrives on top of a track, to the time the target sector passes directly under the head and can be read / written
Transfer rate:	5 Mbytes / sec	transfer rate for data stored on contiguous sectors of the same track (does not include seek, latency times)

Now consider a 4MB file stored in two identical hard disks.

Disk A was initially empty. Therefore Windows was able to store the entire file on adjacent disk clusters. The file ended up occupying exactly 8 tracks on Disk A.

Task I: Calculate the total time required to load the entire file from Disk A into RAM.

Disk B had a badly fragmented file system that had no contiguous free block larger than 20KB. Windows was therefore forced to break the file into several pieces of size 20KB each. The file ended up fragmented into 100 non-adjacent tracks. Each track contains an average of two 20KB file pieces, stored in different (non-contiguous) parts of the track.

Task II: Calculate the total time required to load the entire file from Disk B into RAM. What is the performance reduction caused by disk fragmentation?

PROBLEM 5: Web Home Page Construction (30%)

In this problem, you will construct a simple personal web site, consisting of *at least* two connected web pages: "Home" and some other page. Your home page, which can be kept simple, must include, at the minimum:

- A greeting
- A short paragraph explaining its purpose
- Some image / picture / graphic (e.g. your photo)
- A list of at least three study or work related links
- At least one fun or extracurricular related link
- A clearly labeled direct link to your second web page
- A Java applet from the web site.

> ☝ Your web page will be accessible to the whole world! Therefore ***you should apply appropriate discretion when choosing the content of your page***.
> If you do not wish to publish anything about yourself on the home page, you can also choose an alter ego for this assignment.

Detailed information on constructing and publishing your home page can be found on a teaching note, . The teaching note will also explain how to insert your photo and include links to the web sites. Having created your web site with home and additional pages be sure to publish it on the server, and access it from a browser to make sure it works as you intended.

The only thing you need to submit for this assignment is the URL of your home page.

IMPORTANT NOTE: If you already have a home page that satisfies the above minimum requirements, you may simply submit the URL of your existing home page. Also, feel free to develop your page using whatever tools you are most comfortable with. Finally, you may publish your page on any server you have access to .