Problem 1 +1

Q 1-1: *Name 6 technological advances in hardware and the year in which they were invented (for one of the advances you can mention the decade only), in the early stages of computing (late 40s and 50s), which helped in the advancement of computer hardware designs.*

A 1-1:

Year	Hardware Advance
1947	Transistor
1950s	Printed circuit board
1953	Ferrite core memory
1954	Silicon transistor
1956	Hard disk storage
1958	Integrated circuit

Q 1-2: *Name all the microprocessors mentioned in the hardware history document (name at least 5), the year they were introduced and the computer name or type that they were used for originally.*

A 1-2:

Processor	Year	Computer
Intel 4004	1971	Desktop calculator
Intel 8080	1975	MITS Altair 8800
Zilog Z-80	1977	TRS-80
Motorola 6502	1977	Apple II
Intel 8088	1981	IBM PC
Motorola 68000	1984	Apple Macintosh

Q 1-3: *outline 5 Mainframe computers, the name of the company that introduced them, the year introduced (or the approximate period), and some narrative about the design and intended usage.*

A 1-3:

Mainframe	Year	Company	Details
7090	1959	IBM	First all transistor system. Used by the Air Force in their Defense Early Warning system.
360	1960	IBM	Consolidation of 5 of IBM's models to create a complete system.
CDC 660	1964	Control Data Corporation	Considered the first supercomputer. Operated at 9 Mflops.

ILLIAC IV	1964	Burroughs	First computer to use a parallel design. Had 64 identical computers operating in parallel.
Cray-1	1976	Cray Research	Cylindrical to shorten distances between components. Provided for 100 Mflops.

Q 1-4: *the UNIVAC-1 was introduced in 1954. Using figure 1 in one of the supplied articles identify the expected Floating Point Operations (FLOPS) rate (give an approximate range) of a computer designed in 2004.*

A 1-4: 10^{14}-10^{15} FLOPS

Problem 2

+1

Q 2-1: *Read the development of the Operating Systems Windows, and UNIX during the 1980's. Discuss the reasons Windows was developed and why it caught on so fast. Who was the primary user of Windows in the 1980s? Why was UNIX still being advanced and who was the primary user of UNIX in the 1980s? List the major events that affected UNIX in the period 1980-1990 (other than simple releases) and the dates of those events.*

A 2-1: Windows was developed because it met the needs of the users who were looking for a single-user OS with applications to solve typical business needs. Windows caught on so fast because it offered an easy-to-use GUI, device independent graphics, and multitasking support. It was also cheap and ran on IBM compatible PCs, which were widely used in that era and now. The primary user of Windows in the 1980s was a home/business user.

UNIX was still being advanced in order to support changes in hardware (memory, I/O devices, etc.). It was also being advanced to support new theories about OS structure, such as the micro-kernel. The primary user of UNIX in the 1980s would have been mainly academia and research groups. There was still debate about UNIX being commercially viable[1] in the mid 1980s.

Year	Event
1980	4.0BSD released in October
1983	SCO delivers its first packaged UNIX system called SCO XENIX System V for Intel 8086 and 8088 processor-based PCs
1985	The GNU manifesto is published in Dr. Dobb's Journal

[1] http://www.unix.org/what_is_unix/history_timeline.html

1987	Sun and AT&T lay the groundwork for business computing in the next decade with an alliance to develop UNIX System V Release 4
1989	SCO ships SCO UNIX System V/386, the first volume commercial product licensed by AT&T to use the UNIX System trademark.

Table 1: Major UNIX events of the 1980s

Q 2-2: *by doing some research on the Internet identify the goals of the GNU manifesto. Give a short history of its development and purpose. Identify its differences over "normal" UNIX. The inventor was Richard Stallman, a researcher who originally worked at MIT. Limit your answer to about half of a typed page.*

A 2-2: Goals:

- Write a complete, UNIX-compatible software system to be given away for free for everyone to use
- Permit everyone to modify and redistribute GNU
- Ensure that all future versions of GNU are free, by prohibiting distributors from restricting further redistribution
- Make proprietary software obsolete

In the 1970s, Richard Stallman worked in a group at MIT that exclusively used free software, often provided directly from vendors. Programmers were free – as in freedom – to communicate and share these programs and changes to these programs with each other.

By the 1980s, the times had changed and most vendors were distributing proprietary code, which prohibited sharing. Stallman did not appreciate this trend and started the GNU Project with the first item to be implemented being the operating system itself. Starting from this point, all software on a computer can actually be free.

Over the course of many years, the GNU operating system took shape through the implementation of many of the standard UNIX services, system programs, games, office suites, etc. In 1992, the last major hurdle to providing a complete and free operating system was jumped when Linux was combined with GNU; thus, the GNU/Linux operating system.

GNU is different from UNIX in a couple of ways. For starters, there is no corporate entity with control over GNU as there is over UNIX (currently The Open Group).

Additionally, users and developers of GNU are free to do whatever they like with the operating system: modify it, repackage and distribute it, call it something else, etc. as long as the source code is provided and free and open (the software license is not restricted in any way). A user or developer is even free to charge for the

service of distributing the software or providing software support, but cannot charge for the software itself.

Problem 3 +1

Q 3-1: Exercise 1.5 from text: *Describe the difference between symmetric and asymmetric multi-processing. What are the three advantages and one disadvantage of multiprocessor systems?*

A 3-1-a: Symmetric multiprocessing requires 2 or more processors, each running an identical operating system. Essentially, each processor is its own master.

Asymmetric multiprocessing requires 2 or more processors, with all but one being slaves to a master processor, which schedules the processes that all other processors will run.

A 3-1-b:

Three advantages:

1. Increased throughput – more work in less time.
2. Economy-of-scale – systems can cost less due to less hardware redundancy.
3. Increased reliability – The system can still, possibly, function even when one processor goes down.

Disadvantage:

1. One processor may sit idle while another is overloaded.

Q 3-2: Exercise 2.3 from text: *Describe three general methods for passing parameters to the operating system.*

A 3-2:

1. Directly to register(s)
2. Stored in a block (table) in memory. Just the address of that block is passed in the register
3. Pushed onto the stack and popped off by the OS

Problem 4 +1

Q 4-1: Exercise 2.6 from text: *What are the advantages and disadvantages of using the same system-call interface for manipulating both files and devices?*

A 4-1:

Advantages:

- The interface can be generic; file vs. device neutral. Thus, a developer only needs to develop against one interface.
- Single-point-of-failure in regards to bugs. All related code is in a single place; therefore, one fix has an effect on many resources.
- New types of file-systems or devices can be added without requiring a change to the operating system.

Disadvantages:

- It may not be immediately obvious to a developer what type of resource is being utilized.
- A bug in the code has the potential to affect all devices and file access.

Q 4-2: Exercise 2.7 from text: *What is the purpose of the command interpreter? Why is it usually separate from the kernel? Would it be possible for the user to develop a new command interpreter using the system-call interface provided by the operating system?*

A 4-2: The purpose of the command interpreter is to allow users to interface with the OS by obtaining and executing the sequence of user-specified commands. The command interpreter is usually separate from the kernel because it runs in user-mode; it is needed by users, not the kernel. On many systems, the user has a choice of what command interpreter (shell) to use. Often times, the command interpreter does not even recognize a command, but instead defers to a system program specified in the command.

It is most definitely possible for a user to develop a command interpreter that uses the operating system's call interface. Many interpreters do this already.

Problem 5 +1

Q 5-1: Exercise 2.10 from text: *Why does Java provide the ability to call, from a Java program, native methods that are written in, say, C or C++? Provide an example of a situation in which a native method is useful.*

A 5-1: Java's goal is to allow developers to write code once, compile it once, but run it "anywhere". In order to do so, Java provides a virtual machine which separates Java code from the underlying OS.

Most of the Java API is actually written in Java yet some of those APIs provide support for file handling, process control, network I/O, etc. These functions are OS dependent and as such, Java needs a way to implement this functionality per OS. Thus, the JNI (Java Native Interface) has been introduced allowing Java, and any other Java developer, to tie "generic" Java code to native OS functionality.

A typical example of the need for a native method would be to remove a file from the file-system via the *java.io.File.delete()* method.

Q 5-2: Exercise 2.14 from text: *What is the main advantage for an OS designer of using virtual-machine architecture? What is the main advantage for a user?*

> **A 5-2:** In both cases, the designer and user, the main advantage is isolation. From an OS designer's viewpoint, a VM enables the designer to modify an OS without disrupting normal system operation. A designer can also test an OS under various conditions or with many types of devices without actually requiring the physical presence of those conditions or devices.
>
> From a user's viewpoint, a user does not have to share any resources. The user is isolated from the effects of multiple users or programs demanding attention from various files or devices.

Problem 6

The solution to this problem can be found in two places:

1. On the CS450 server, [cs450.cs.iit.edu:~knowlden/assignment1/problem6]
2. In the same archive as this document was found under the directory named [problem6]

The script [calc.src] was actually written using the TDD (Test Driven Development) methodology. In TDD, no code – in this case, commands in the script [calc.src] – is written until a test has been written to assert the expected behavior. The artifact created from this process is a test file – in this case, the script named [testcalc.src].

Additionally, a directory named [.backup], writable only by my user, has been created in the [~knowlden/assignment1/problem6] directory in the event that something unforeseen happens to my scripts.

Problem 7

The solution to this problem can be found in two places:

1. On the CS450 server, [cs450.cs.iit.edu:~knowlden/assignment1/problem7]
2. In the same archive as this document was found under the directory named [problem7]

The script [greptests.src] was actually written using the TDD (Test Driven Development) methodology. In TDD, no code – in this case, commands in the script [greptests.src] – is written until a test has been written to assert the expected behavior. The artifact created from this process is a test file – in this case, the script named [test greptests.src].

Additionally, a directory named [.backup], writable only by my user, has been created in the [~knowlden/assignment1/problem7] directory in the event that something unforeseen happens to my scripts.