Author: Mike Slone
SID: 10213953
Assignment 1
Due: September 11, 2005

Problem 1

1)1940's

-Bell Telephone Labratories develops transistor
-ENIAC is developed by Mauchly and Eckert

+0.25

1950's

-UNIVAC computer is developed which can store 12,000 digits
-Texas Instruements and Fairchild Semiconductor both announce the integrated circuit
-MIT researchers build TX-0, the first general-purpose, programmable computer built with transistors
-Integrated Circuit patent applied for by Texas Instruments

2)

Intel 4004	1971	"desktop calcluator"
Intel 8080	1975	MITS Altair 8800
Zilog Z-80	1977	TRS-80
Motorola 6502	1977	Apple II
Intel 8088	1981	"personal computer

+0.25

3)

1) In 1964, CDC introduced the CDC 6600. It was the first "supercomputer" to be a commercial success. It operated at 9 Mflops. It was developed by Seymour Cray.

2) In the 60's, the ILLIAC IV was developed by Burroughs. First computer to use a parallel (non-von neumann) design. It had 64 identical scalar computers working in parallel. + 0.25

3) In 1968, another record was set for speed with the 40-Mflop CDC 7600.

4) In 1968, Burroughs released the B2500 and B3500, the first mainframes to

incoroprate integrated circuits as opposed to discrete components.

5) In 1970, CDC released the "STAR" which had a vectorial, non-von neumann architecture for the parallel processing of vectors as opposed to scalar values.

4) According to "Figure 1" in 2004, the expected FLOP rate/sec would be about 10^14, or one hundred trillion FLOPS/second.

+0.25

Problem 2

1) Some say that Windows was developed as a platform upon which the Microsoft Office products could be used. Microsoft Excel and Microsoft Word for Windows executed from the DOS prompt, but ran a run-time version of Windows in which the programs ran. The beginning of the revolution came, however, when Aldus Pagemaker was released for Windows. This was the first significant non-Microsoft product to run in Windows. The popularity of Windows spawned from the openness with which other development firms could create software and hardware to work with it. The next step came with Windows 3.0, when the Virtual Machine architecture and overlapping window structure allowed multiple programs to be simultaneously run smoothly. UNIX kept advancing, as the appeal was to a completely different market. UNIX was still much more useful for networking and mainframe work than Windows was. Windows had appeal to the individual user (business or home user), whereas UNIX was aimed at networking purposes. One of the largest effects on the progress of UNIX was the development of the GNU Public License, which was introduced in 1985 (see below). This allowed for the development of open-source products without the worry of corporate acquisition. The ninth and tenth versions of UNIX also came out during the 80's allowing for advancement in the capabilities for both users and developers. Another big step for UNIX was when UNIX V/386 became the first volume commercial product licensed by AT&T to use the UNIX trademark in 1989—marking UNIX as an entity independent in importance of AT&T. +0.5

2) GNU Public License (GPL) came from the philosophy of Richard Stallman and the GNU Project. He was inspired by his experience with James Gosling. Gosling was the creator of the Java programming language, NeWs, and UniPress, over Emacs. Gosling took Stallman's Emacs, and created the first C-based version running on Unix in 1982. Gosling originally allowed the code to be open-source, and Stallman used this code in the first version of GNU Emacs in 1985. Gosling later sold his code to UniPress who forced Stallman to cease distribution of GNU Emacs. This inspired Stallman to prevent future open-source properties from being exploited in such a way, Stallman created the GNU Public License, a license to cover and protect open-source software projects from corporate acquisition. The differences over "normal" UNIX, is that GPL is based on open-source architecture, where many developers can work from each other's progress and take pieces of software in multiple directions. It also allows developers to learn from one-another, since GNU developers are not sworn to secrecy, and are in fact encouraged greatly to share ideas. Developers can look at what's been done, make suggestions for improvement, and take the next step. +0.5

Problem 3

1.5) In asymmetric processing, multiple "slave" processors are scheduled by a "master" processor or are preassigned tasks, each having a specific agenda. In symmetric processing, each processor is on the same level as all the others, and operates within the operating system (Solaris is an operating system which supports this). ~~The advantage of Symmetric multiprocessing is that each processor can be dedicated to a single process, thus allowing multitasking without a deterioration of efficiency. The advantage of asymmetric multiprocessing is that there are fewer problems with workload distribution~~ +0.2

~~(for instance, one processor may be handling a much more hefty task than the others, but the others are still not helping).~~

2.3) ~~Parameters can be passed to the operating system thorough the shell, through system calls, and through a graphical user interface.~~

Problem 4

2.6) The advantage of using the same system-call interface for devices as files is that you can learn one and not need to know much more to use the other. The disadvantage is that the systems are in fact a little different, and misuse can lead to user-invoked problems both with hardware and with file systems. +0.3

2.7) The command interpreter takes a command from the shell and breaks it down in such a way that the operating system can understand. It takes the information, breaks it down, interprets it, and sets the resulting action in motion. It is separate from the kernel because the kernel is the platform of execution, whereas the command interpreter is the bridge between that platform and the user. This allows a developer to change the way the user and platform of execution are bridged. It would be possible for a developer to create a new command interpreter using the system-call interface provided by the operating system. +0.5

Problem 5

2.10) Many methods work much quicker when not run on a virtual machine, so therefore, some Java programs may benefit from calling native methods written in C or C++. One example of an advantageous use for this would be an operation which needs to manipulate large amounts of data in memory efficiently. Since Java relies on a built-in garbage collector, a programmer may be able to provide more efficient garbage collection routine based on knowledge of the data being manipulated. +0.5

2.14) Using a virtual machine architecture allows the operating system to run multiple processes simultaneously while creating the illusion that each has its own full set of dedicated resources. This is advantageous to the operating system developer, as one piece of software can run nearly identically on different machines, without knowing beforehand what sorts of resources will be at hand. The advantage to the user, of course, is that they can change hardware and software without worrying about how they relate to one-another. + 0.5