HW # 2: Theme: Binary number System (Review prerequisites)

1.   Name all eight 32-bit general-purpose registers. What is the general function of each of the registers? Which of these registers can be addressed in parts?

EAX – accumulator; can be addressed in parts

EBX – base register ("for pointers"); can be addressed in parts

ECX - loop counter; can be addressed in parts

EDX – data register (used in conjunction with accumulator); can be addressed in parts

EBP - extended frame pointer (stack base pointer)

ESP - stack pointer

ESI - source index register

EDI - destination index register

2.   What do the Sign Flag, Zero Flag, Auxiliary Carry Flag, and Parity Flag indicate when set?

Sign Flag – Indicates a negative number

Zero Flag – Indicates a value of 0

Auxiliary Carry Flag – Indicates a carry from bit 3 to bit 4

Parity Flag – Indicates Even Parity

3.   What do the Overflow Flag and Carry Flag indicate when set? How do they differ?

Overflow Flag – Indicates a MSB flip other than by explicit sign flip (in signed arithmetic)

Carry Flag – Indicates a value requiring more bits than are available in result register (in unsigned arithmetic)

4.   Detail the process by which instructions and data are read from memory.

When memory must be read, the memory controller places the appropriate address on the address bus. A read flag switches to active low, and the memory controller places the instruction or data on the data bus, where it may then be read by the CPU following a switch to active high by the read flag.

5.   How is multitasking achieved?

The processor (if it supports task switching) iterates over the set of all concurrent tasks in a given precedence, and at a time interval set by a scheduler utility. To a human user, it appears as if the processes are being run simultaneously. An alternate method is to multi-thread multiple processes within the SAME program, thereby achieving multitasking.

7.   What do you understand by Real-address mode, Protected mode, Multi-segment model, and Paging? How does Real-Address mode differ from Protected mode? What are the defining attributes of a Multi-segment model? How does paging benefit a system? ***Discuss in detail.***

Real-Address Mode allows memory to be accessed explicitly and without restriction, and as such, can cause a system crash if not used properly. Protected mode, conversely, allots separate segments of memory to each program, and denies access to incorrect segments. Protected mode is the normal running mode of x86 machines. The Multi-Segment Model refers to a system in which each program has a distinct segment of memory of a certain (non-standard) size. Paging refers to the process by which a computer, needing more memory than is currently available, splits data into 4 kb "pages" and temporarily stores pages that are not being used on the disk until they are needed, giving the illusion of more available memory.