X86 ARCHITECTURE

Basic History

Assembly language programming
By xorpd

xorpd.net

OBJECTIVES

- You will learn about the first processors in the x86 family.
- You will be introduced to the concept of backwards compatibility as it applies to the x86 processors family.
- You will know about some of the different modes being used in x86 processors.

QUICK REMINDER ABOUT PROCESSORS

- The processor is the heart of a modern computer.
- A processor is a electrical component that has the following functions:
  - Read instructions.
  - Execute instructions.
    - Arithmetic calculations. (Like adding, subtracting and multiplying).
    - Reading / Writing memory.
    - Input / Output.
- The instructions to be executed are written ahead of time, in memory.
  - The collection of instructions in memory is called a program.

BASIC HISTORY OF X86 ARCHITECTURE

- x86 is a general name for a family of processors.
  - All of those processors can understand (more or less) the same program instructions.
- Intel introduced the 8086 processor in 1978.
  - The first processor in the x86 family. (Based on intel's 8080 8 bit processor).
  - Primarily developed for embedded systems (Small special purpose computers), and other small single user computers.
  - Has 16 bits support.

The 8086 processor
http://en.wikipedia.org/wiki/File:KL_Intel_D8086.jpg

BASIC HISTORY (CONT.)

- Intel's 80386 processor was the first in the series to be 32 bit compatible (Introduced in 1985).
- In 2003 AMD introduced Opteron – The first x86 compatible processor with 64 bit extension.
  - Intel adopted AMD's architectural extensions into its next processors designs.

http://en.wikipedia.org/wiki/File:KL_Intel_i386DX.jpg

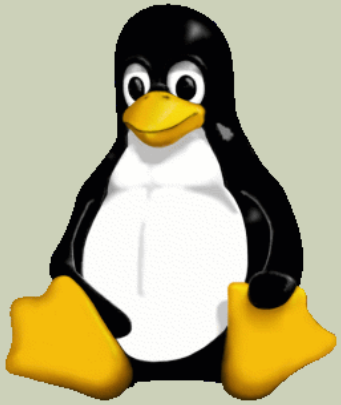
http://en.wikipedia.org/wiki/File:AMD_Opteron_2212_IMGP1795.jpg

BASIC HISTORY (CONT.)

- Today (2014) x86 based processors are very common in stationary and portable computers.
  - x86 processors are produced by many different companies.
  - Many stationary computers and laptop ship with an x86 based processor.
  - Most operation systems can “run” on x86 processors.

Windows

Mac OSX



Linux

BACKWARDS COMPATIBILITY

- Backwards compatibility: Programs that work on old processors will work on newer processors.
  - Easier for developers to upgrade their programs.
- Backwards compatibility was mostly preserved since the 8086 processor.
  - Programs that run in 1978 on the 8086 processor can (basically*) run today on the newest x86 processors!
- x86 backwards compatibility is said to be a major factor in the widespread adoption of x86 processors today.
- Backwards compatibility sometimes forces modern x86 processor designers to take strange design choices.

X86 PROCESSOR MODES

- X86 Processors have several modes, that help with the backwards compatibility.
  - New x86 processors work just like older x86 processors, but they have more operation modes.
- **Real mode** is the operation mode of the 8086 processor.
  - Very limited memory access. (Access to less than 1MB of code and data).
  - Every x86 processor starts in this mode.
- **Protected mode** (Since intel's 80286 processor).
  - Access to larger amounts of memory. (1 GB in 16 bit).
  - Provides memory protection. Helps the operation system manage a multitasking environment.
  - This course will mostly deal with the Protected Mode.
- **Long mode** (Since the AMD's Opteron processor)
  - Can handle data of size 64 bits.
- There are more modes :)

WE ARE LEARNING BACKWARDS

- The long history of x86 processors could make them seem a bit complicated at first sight.
- We are going to learn from the end: Protected mode (and Long mode) are easier to deal with in the beginning.
- The operation system deals with most of the earlier modes of the processor for you.
  - Most of the time, you will be programming in the Protected mode or Long mode.
  - Usually you are not going to deal with real mode, unless you want to write an operation system or a BIOS.
- Learning to program in more recent modes (Protected, Long) will give you a strong basis to approach real mode (and other modes) later, if you want to.