X86 MEMORY MANAGEMENT

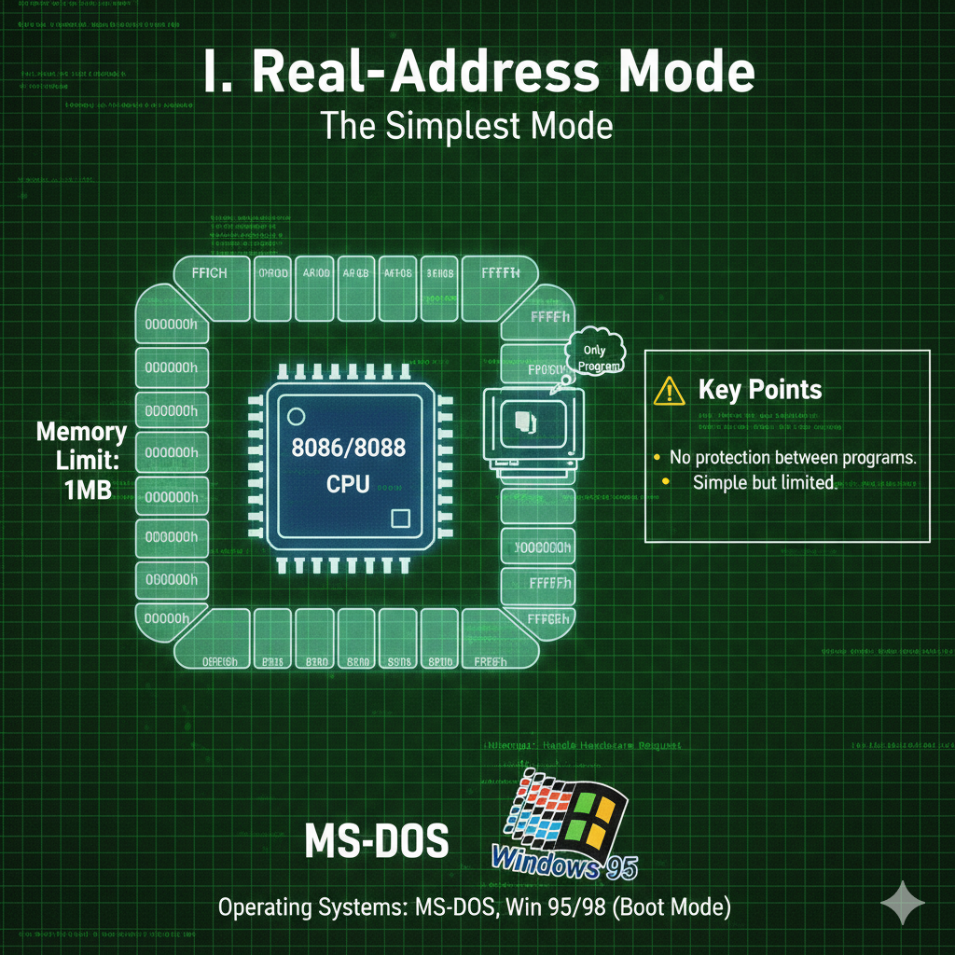
x86 processors manage memory differently depending on the mode of operation. These modes determine how programs access memory and system hardware.

I. Real-Address Mode

* The simplest mode.
* Memory limit: **1 MB** (from 00000h to FFFFFh).
* Only **one program runs at a time**, but the processor can pause it briefly to handle hardware requests called **interrupts**.
* Programs can access **any memory location**, including memory mapped to hardware.
* Operating systems using this mode: **MS-DOS**.
* Windows 95 and 98 can boot into this mode.

**Key points:**

* No protection between programs.
* Simple but limited in memory and multitasking.

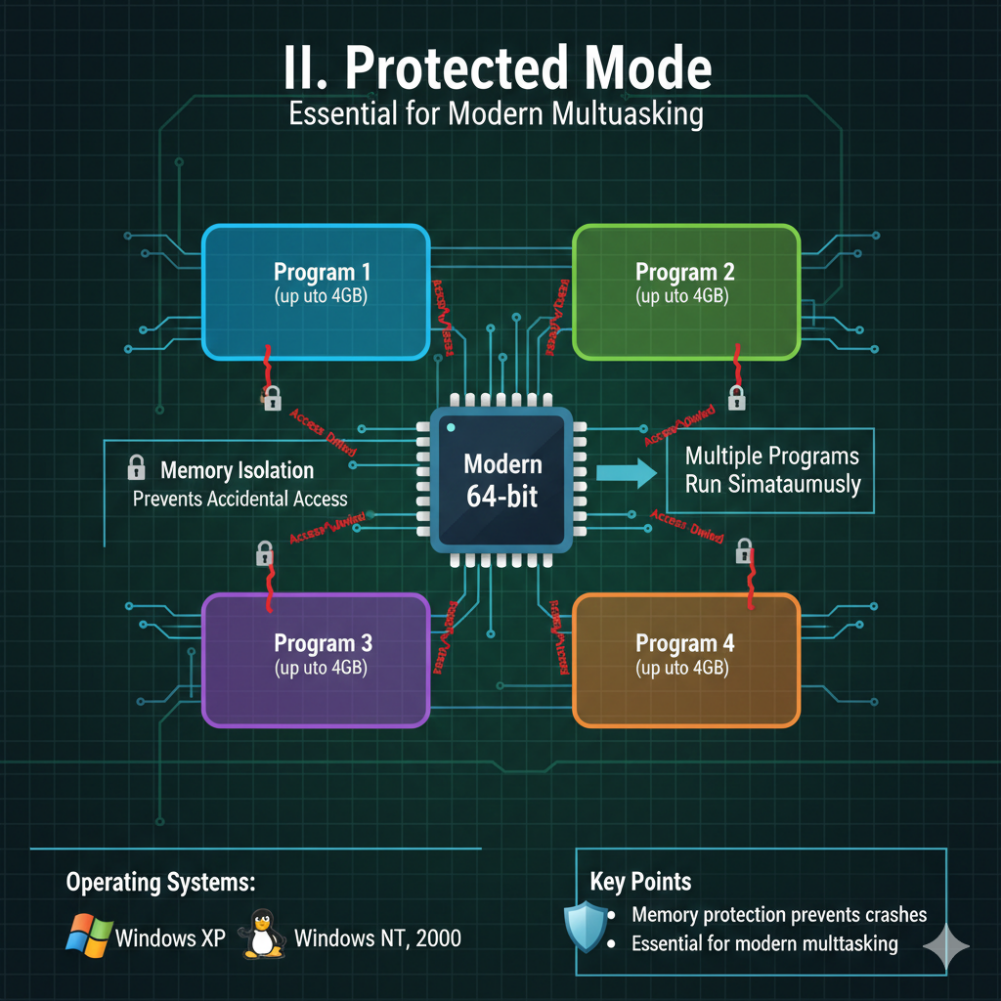


II. Protected Mode

* Allows multiple programs to run **simultaneously**.
* Each program gets **up to 4 GB of memory**, isolated from other programs.
* Prevents accidental access to other program’s code or data.
* Operating systems using this mode: **Windows NT, 2000, XP, Linux**.

Key Points:

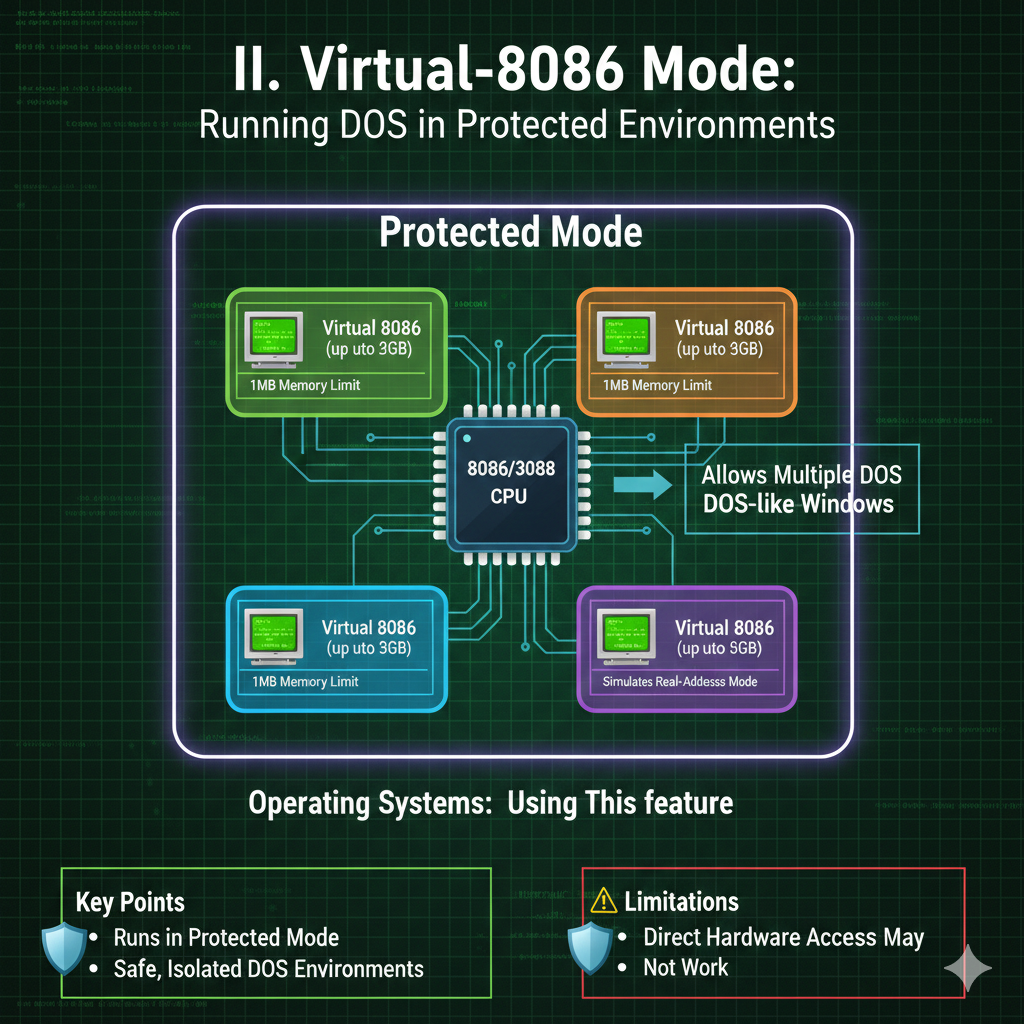
* Memory protection prevents crashes caused by one program affecting others.
* Essential for modern multitasking operating systems.



III. Virtual-8086 Mode

* Runs in **protected mode** but creates a **virtual 8086 machine**.
* Each virtual machine has **1 MB memory**, simulating a real-address mode environment.
* Allows multiple DOS-like windows to run safely.
* Operating systems using this feature: **Windows NT, 2000, XP**.

**Limitations:**Programs that directly access hardware may **not work properly** in this mode.

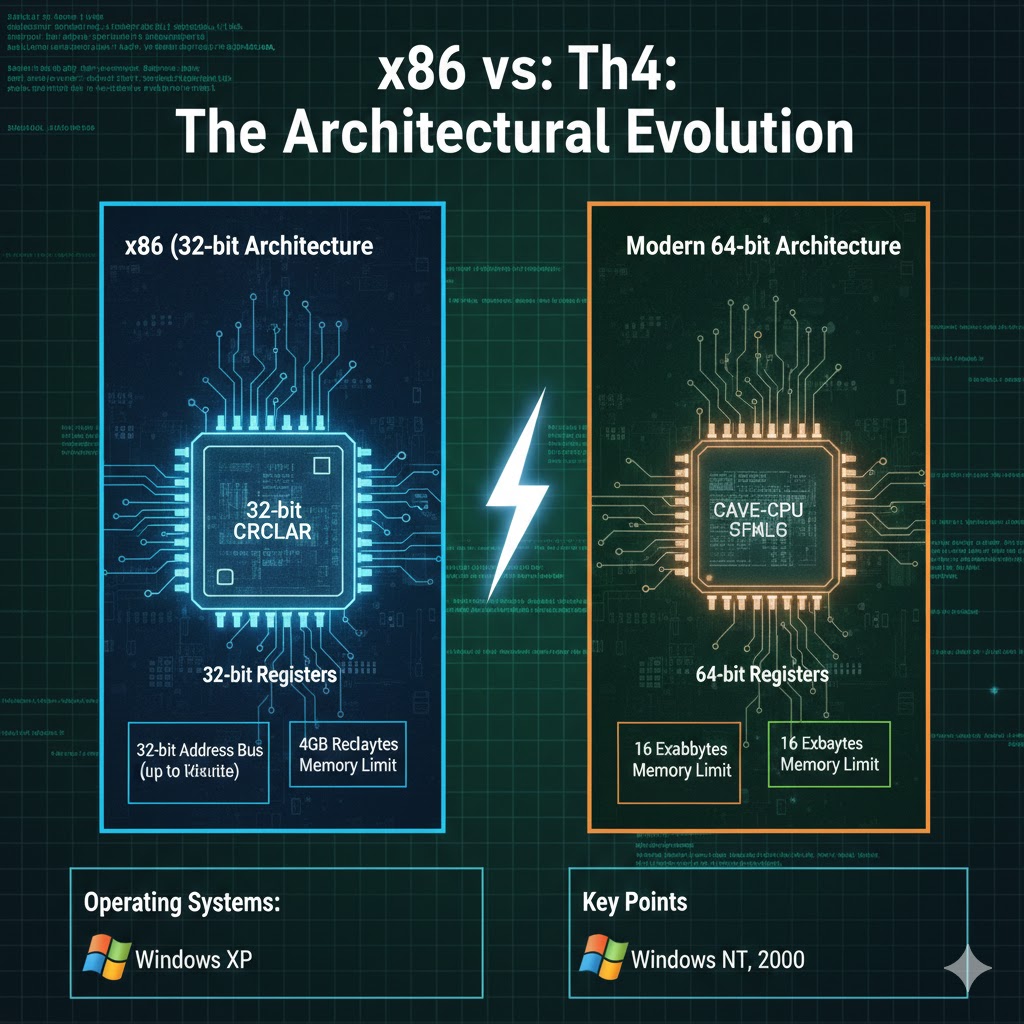


X86-64 PROCESSORS VS X86 PROCESSORS

x86-64 is an **extension** of the x86 instruction set that supports **64-bit computing**.

I. Main Features Of x86-64

* Uses **64-bit addresses**, allowing a huge virtual memory space: 2^64 bytes.
* Physical memory support: **up to 256 TB**.
* **16 general-purpose registers** (8 more than x86).
* **64-bit instruction operands**.
* Does **not support** 16-bit real mode or virtual-8086 mode in native 64-bit mode.
* Backward compatible with x86 instructions.



II. Modes In Intel 64 Architecture

**Compatibility Mode**: Runs older **16-bit and 32-bit programs** without recompiling.

**64-bit Mode**: Native mode for modern 64-bit applications.

III. Registers

In modern 64-bit CPUs, there are several types of registers you need to know, as we’ve discussed in the previous chapter.

There are **16 general-purpose registers**, each 64 bits wide, used for most calculations, addressing, and data storage.

The CPU also has **8 floating-point registers**, each 80 bits wide, which store floating-point numbers and intermediate results for precise math.

The **RFLAGS register** is 64 bits, holding status flags that report the results of operations and control CPU behavior.

Only the lower 32 bits are used in most instructions. The **RIP register** is the 64-bit instruction pointer, keeping track of the next instruction to execute.

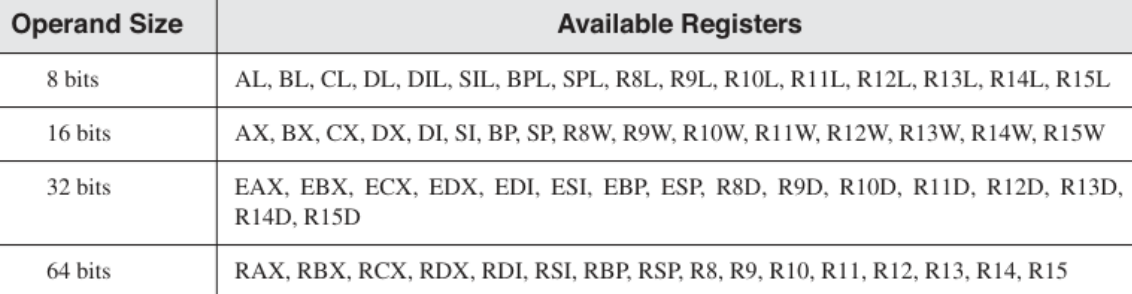
For multimedia and SIMD tasks, there are:

* **8 MMX registers** (64-bit), mostly legacy
* **16 XMM registers** (128-bit), used for SSE, AVX, and other SIMD extensions

The **REX prefix** in 64-bit mode allows instructions to access the full 64-bit registers.

Some important notes about operands and sizes:

* Registers can operate on **8, 16, 32, or 64-bit values**
* In 64-bit mode, the **default operand size is 32-bit**, unless you override it with prefixes or instruction modifiers



64-BIT REGISTER QUIRKS

I. THE REX PREFIX LIMITATION

In **64-bit mode** (also called **Long Mode**), the CPU introduces **new registers** (R8–R15 and their byte versions) but also enforces a strict rule when accessing **byte-sized registers**.

There’s an important rule when working with registers in modern x86-64 CPUs.

You **cannot mix the old high-byte registers**—AH, BH, CH, or DH—with the **new low-byte registers**, like SIL, DIL, BPL, SPL, or R8B through R15B, in the same instruction.

In other words, a single instruction can’t operate on both an old high-byte register and one of the new low-byte registers at the same time.

This is just a hardware limitation, so you have to plan your instructions accordingly.

Think of it like trying to mix oil and water—they exist in the same CPU, but they **don’t combine in a single operation**.

**Why?**

* To access the new registers, instructions use a special **prefix code** called **REX**.
* The CPU hardware is designed so that **REX cannot coexist with legacy high-byte registers** in the same instruction.
* Trying to mix them will **cause an invalid instruction** error.

**Examples (HTML 0017 in this folder):**



x86 HARDWARE ARCHITECTURE

4.0 Introduction: The Anatomy of the Machine

If the CPU is the "Brain," the Hardware Architecture is the "Body." You cannot become a master programmer or reverse engineer without understanding the physical terrain your code travels through.

4.1 The Motherboard: The Central Nervous System

The **Motherboard** (or Mainboard) is a complex printed circuit board (PCB) that acts as the hub for all communication.

**4.1.1 The Bus**

The "Bus" is not a vehicle; it is a set of microscopic copper wires etched onto the motherboard.

* **Function:** It is the highway system. Data travels along these wires between components.
* **Width:** If a bus is "64-bits wide," imagine a highway with 64 lanes. 64 bits of data can travel simultaneously.

**4.1.2 Key Components**

1. **CPU Socket:** The "Throne." It houses the processor. Different CPUs require different socket shapes (e.g., LGA 1700 for Intel, AM5 for AMD).
2. **Memory Slots (DIMM):** Where the RAM sticks live.
3. **Expansion Slots (PCIe):** Parking spots for high-speed add-ons like Graphics Cards (GPUs) or WiFi cards.
4. **The BIOS Chip:** A non-volatile chip that holds the startup code.

**4.2 THE CHIPSET: The Traffic Controllers**

The CPU is too fast and too important to talk to a slow USB keyboard directly. It delegates this work to the **Chipset**.

**4.2.1 The Memory Controller Hub (MCH)**

Historically called the **Northbridge**.

* **Job:** High-speed logistics. It connects the CPU to the **RAM** and the **Graphics Card**.
* **Performance:** Intel’s "Fast Memory Access" technology allows the MCH to look ahead at memory requests and reorder them for maximum speed (like a chef arranging orders to cook efficiently).
* *Modern Note:* In modern CPUs (i3/i5/i7/Ryzen), the MCH is moved *inside* the CPU chip itself for speed.

**4.2.2 The I/O Controller Hub (ICH)**

Historically called the **Southbridge**.

* **Job:** Slow-speed logistics. It handles USB, Hard Drives, Audio, and Keyboards.

**4.3 MEMORY HIERARCHY: Types of RAM & ROM**

Not all memory is created equal. We categorize memory based on **Permanence** (Does it keep data when power is off?) and **Speed**.

**4.3.1 ROM (Read-Only Memory)**

* **Nature:** "Written in Stone."
* **Permanence:** Non-Volatile. Data stays forever.
* **Usage:** Storing the **BIOS/UEFI** firmware. This is the first code that runs when you press the power button.
* *Security Insight:* "Rootkits" that infect the BIOS are terrifying because formatting the hard drive does not fix them. They live in the ROM chip.

**4.3.2 EPROM (Erasable Programmable ROM)**

* **Nature:** "The Whiteboard."
* **Mechanism:** It can be erased, but it requires blasting the chip with distinct **Ultraviolet (UV) Light** through a small quartz window on the chip.
* **Usage:** Old embedded systems. Rarely used in PCs today (replaced by EEPROM/Flash, which can be erased via electricity).

**4.3.3 DRAM (Dynamic RAM)**

* **Nature:** "The Leaky Bucket."
* **Mechanism:** It stores data using tiny capacitors. These capacitors leak electricity and lose the data within milliseconds.
* **The Refresh Cycle:** To keep the data alive, the memory controller must "recharge" (refresh) every row of RAM thousands of times per second.
* **Usage:** This is your **Main System Memory** (e.g., 16GB DDR4). It is cheap and dense.
* *Security Insight:* **Cold Boot Attacks.** If you freeze a DRAM stick with compressed air, the "leak" slows down. Hackers can pull the RAM out of a running laptop, freeze it, and read encryption keys from it minutes later on another machine.

**4.3.4 SRAM (Static RAM)**

* **Nature:** "The Flip-Flop."
* **Mechanism:** Uses a circuit of 6 transistors to hold a bit. It does **not** need refreshing. As long as power is on, data is rock solid.
* **Speed:** Blazingly fast. Much faster than DRAM.
* **Usage:** **CPU L1/L2/L3 Cache.** It is too expensive and bulky to use for main memory.

**4.3.5 VRAM (Video RAM)**

* **Nature:** "The Dual-Ported Artist."
* **Mechanism:** It is **Dual-Ported**. This means two devices can access it at the exact same time.
  1. The **GPU** writes new frames to it.
  2. The **Monitor** reads frames from it to display.
* **Usage:** Graphics cards.

**4.3.6 CMOS RAM (Complimentary Metal Oxide Semiconductor)**

* **Nature:** "The Battery-Powered Notepad."
* **Mechanism:** A tiny distinct chunk of memory kept alive by a coin-cell battery (CR2032) on the motherboard.
* **Usage:** Stores settings that must survive a power outage: **System Date/Time** and **BIOS Settings** (Boot order, Overclocking speeds).
* *Hack:* If you forget your BIOS password, removing this battery resets the memory, clearing the password.

**4.4 COMPARISON TABLE: MEMORY TYPES**