History of Linux and the command line

Zubin

January 30, 2026

Contents

1 Compiler vs interpreter

In the C programming language, we use a compiler. To understand the difference between an interpreter and a compiler, we can use an analogy of landing on a planet where inhabitants speak a strange language called "Gobbledygook". To get a mechanic to repair your spaceship, you need a translator.

1.1 The Interpreter

If you choose an interpreter:

- **Process:** The interpreter reads your first instruction, translates it immediately, and the mechanic executes it. Then it reads the second, translates, and executes, and so on.
- **Characteristics:** The interpreter stays with you, translating line by line.
- **Pros:** It allows you to correct mistakes as you go (interactive).
- **Cons:** It is a slow process because the mechanic waits for translation between steps.
- **Etymology:** "Inter" means between. The interpreter is always between your program and the computer.

1.2 The Compiler

If you choose a compiler:

- **Process:** The compiler takes your complete list of instructions and translates the whole lot at once. It then hands the translated list back to you and leaves.
- **Characteristics:** You hand the complete list to the mechanic, who executes them all in one go very quickly.
- **Pros:** Execution is very fast and efficient.
- **Cons:** Takes extra preparation time initially. If there is a mistake, it is too late to fix it during execution.
- **Etymology:** "Compile" means to pile together. It piles together your entire program and translates it all at once.

1.3 Summary

- **Interpreter:** Runs slowly, starts right away, allows you to see how things are going.
- **Compiler:** Takes preparation time, but runs very quickly and efficiently.

2 Memory representation: RAM, cells, word, byte, bit, memory address

How does the computer remember where it has stored the value for a certain variable? To answer this, we need to understand computer memory.

2.1 Types of Memory

One role of the operating system is to manage the computer's memory.

- **RAM (Random Access Memory):** Temporary, volatile memory used to execute programs. It is quick to access.
- **Non-volatile Memory:** Permanent storage, such as the hard drive, used for storing files.

2.2 Structure of RAM

Most programs use RAM during execution.

- **Bit:** A single binary memory cell (0 or 1).
- **Word:** A group of bits forming the fundamental unit of data moved between RAM and the processor.
- **Word Size:** The number of bits in a word (e.g., 8, 16, 32, 64 bits).
- **Byte:** A unit consisting of 8 bits.

2.3 Memory Addressing

To locate data, memory cells are grouped into words, and each word is assigned an address.

- **Memory Address:** A whole number describing the location of a word in memory (similar to house addresses on a street).
- **Example:** If a computer has 8-bit words:
  - Address 0 points to the first word (first 8 bits).
  - Address 1 points to the second word (next 8 bits).

2.4 Memory and C Programming

In C, it is possible to access these memory addresses directly.

- **Access:** You can obtain the address where a variable's value is stored.
- **Optimization:** This allows for low-level memory management and optimization of execution speed.

---

3 Manage the memory with the command line: free, top, htop

To manage memory, we first need to see how much memory is used by the programs running on the Linux system.

3.1 The free Command

The `free` command displays the amount of free and used system memory.

- **Options:** Use `-b` (bytes), `-k` (kilobytes), `-m` (megabytes), or `-g` (gigabytes) to set the unit.
- **Example:** `free -m` displays the table in megabytes.
- **Output:** The `Mem` line shows the total, used, and free memory.

3.2 The top Command

The `top` command shows memory usage per process.

- **Statistics:** The header shows total used and free memory.
- **VSZ (Virtual Size):** Represents the virtual memory for each program (sometimes labeled **VIRT**).
- **Sorting:** Press the **M** key to sort processes by memory usage.
- **Visualization:** Pressing the **S** key (in this specific version) switches to a view showing only memory usage.

3.3 The htop Command

`htop` provides a visual representation of system resources.

- **Visuals:** Bars indicate used and free memory.
- **Sorting:**
    1. Press **F6** to select "Sort by".
    2. Select **M_SIZE** (Memory Size) and press **Enter**.

3. This sorts the list by the **VIRT** column.

Monitoring memory helps identify programs putting too much pressure on the system, which can then be killed if necessary.
