

History of Linux and the command line

Zubin

January 29, 2026

Contents

1 Compiler vs interpreter	2
1.1 The Interpreter	2
1.2 The Compiler	2
1.3 Summary	2

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.

