

## Recitation 3: Unix Timesharing I

### Question

What is UNIX?

**Answer:** UNIX is an operating system, and an operating system is a program that runs on the computer and abstracts away the underlying hardware. UNIX is not used frequently today, but UNIX's architecture, especially its modular design and use of small, specialized tools, has influenced many modern operating systems, such as Linux, macOS, and BSD.

### Question

How is its filesystem designed?

**Answer:** UNIX has a hierarchical filesystem structure, where files and directories are stored in a tree-like structure. Each directory has a parent directory, which is the directory that contains it, and a list of child directories and files. UNIX files can be regular files, directories, symbolic links, or special files (such as device files). Device files are hardware devices like printers or hard drives that are represented as files in the filesystem, allowing interaction through standard file operations. You can access files via pathnames, which can be absolute (starting from the root) or relative (relative to the current working directory).

### Question

Why was it designed to work that way?

**Answer:** The hierarchical structure allows for easy organization and management of files. Users can create subdirectories, organize files as they see fit, and have multiple users accessing the system without conflict. The filesystem was designed with portability in mind, so that the operating system could be easily modified or ported to different hardware architectures, which helped UNIX spread across various systems.

### Question

UNIX was designed for programmers, by programmers. Who was a programmer in this context? How does this affect the way we use computers today?

**Answer:** Unix was primarily created by Ken Thompson and Dennis Ritchie, two researchers and programmers at Bell Labs, which explains the "by programmers" aspect. The "for programmers" part refers to the broader concept of a programmer, meaning anyone who interacts with the computer, even if they aren't directly coding. This philosophy has influenced how we use computers today, emphasizing the importance of tools being user-friendly, with a strong focus on simplicity and flexibility.