CS110: Principles of Computer Systems

Autumn 2021
Jerry Cain
PDF

Lecture 03: Filesystems II, Design Principles

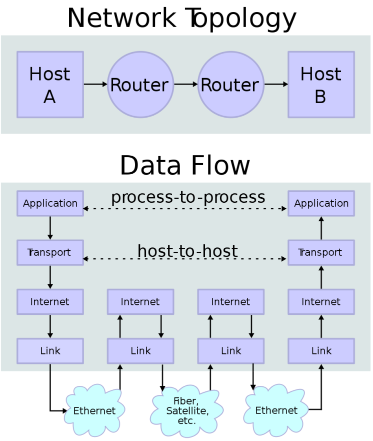
- There are two major design principles relevant to our discussion of filesystems.
    - Modularity and Layering
        - Subdivision of a larger system into a collection of smaller subsystems, which themselves may be further subdivided into even smaller sub-subsystems.
        - Example: **filesystems**, which use a form of modularity called **layering**, which is the organization of several modules that interact in some hierarchical manner, where each layer typically only opens its interface to the module above it.
            - symbolic link layer
            - absolute path name layer
            - path name layer
            - file name layer
            - inode number layer
            - file layer
            - block layer

Lecture 03: Filesystems II, Design Principles

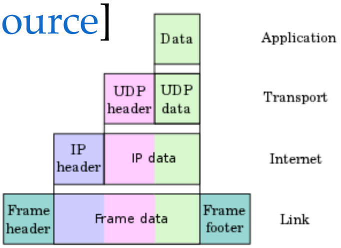
- There are two major design principles relevant to our discussion of filesystems.
    - Modularity and Layering
        - Subdivision of a larger system into a collection of smaller subsystems, which themselves may be further subdivided into even smaller sub-subsystems.
        - Example: **`g++`**, which chains together a series of components in a pipeline (which is another form of layering).
            - the **preprocessor**, which manages **`#include`**s, **`#define`**s, and other preprocessor directives to build a translation unit that is fed to...
            - the **lexer**, which reduces the translation unit down to a stream of tokens fed in sequence to...
            - the **parser**, which groups tokens into syntactically valid constructs then semantically verified by...
            - the **semantic analyzer**, which confirms that the syntactically valid constructs make sense and respect C++'s type system, so that x86 instructions can be emitted by...
            - the **code generato**r, which translate your C++ code into equivalent machine code.

Lecture 03: Filesystems II, Design Principles

- There are two major design principles relevant to our discussion of filesystems.
  - Modularity and Layering
    - Subdivision of a larger system into a collection of smaller subsystems, which themselves may be further subdivided into even smaller sub-subsystems.
    - Example: **computer networks**, which rely on a programming model known as TCP/IP, so named because its two most important protocols (TCP for Transmission Control Protocol, IP for Internet Protocol) were the first to be included in the standard.
      - TCP/IP specifies how data should be packaged, transmitted, routed, and received.
      - The network stack implementation is distributed down through four different layers:
        - application layer (the highest layer in the stack)
        - transport layer
        - network layer
        - link layer (the lowest layer in the stack)
      - We learn application-layer networking in CS110 around Week 7.
      - CS144 teaches all four layers in detail and how each layer interacts with the one beneath it.

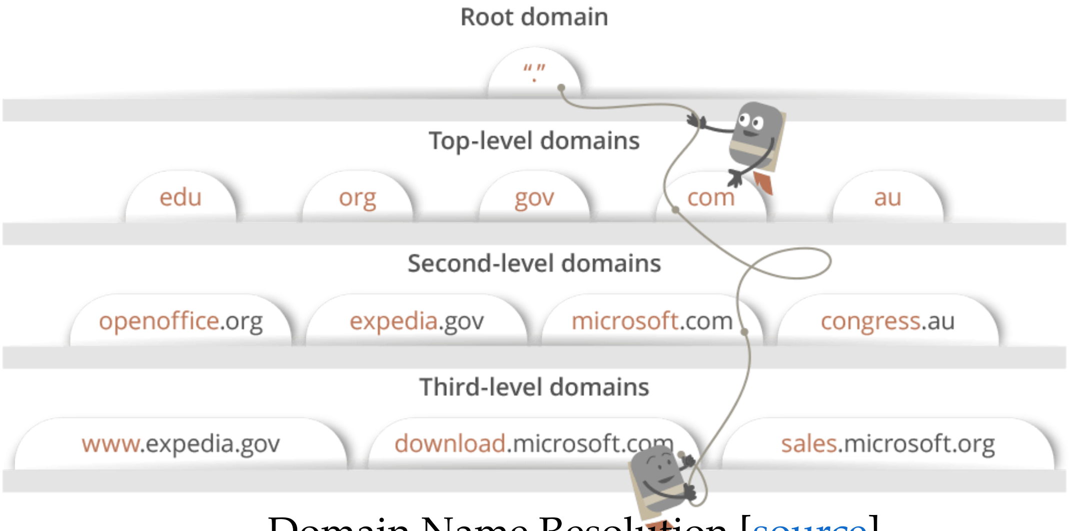


Internet Protocol Suite [source]



Lecture 03: Filesystems II, Design Principles

- There are two major design principles relevant to our discussion of filesystems.
  - Modularity and Layering
  - Naming and Name Resolution
    - Names provide a way to refer to system resources, and name resolution is a means for converting between human-readable names and machine-friendly ones.
    - Examples:
      - Humans prefer absolute and relative pathnames to identify files, and computers work better with inode and block numbers. You'll spend a good amount of energy with Assignment 2 managing the discovery of inode numbers and file block contents given a file's name.
      - Humans prefer names like www.google.com, but computers prefer numbers like 74.125.239.51. The Domain Name System—you might know it as DNS—is a distributed database that maps every domain name to one or more associated IP addresses.



Domain Name Resolution [source]