
High Performance Computing

Lecture 36

Matthew Jacob

Indian Institute of Science

File System Design Issues

1. Disk management: efficient use of disk space
2. **Name management**: how users select files for use
3. **Protection**: of files from users

Name Management

Issues:

- How does a user refer to a file?
- How does the OS locate a file on disk, given its name?
- **Directory**: mapping between file names and file descriptors

Name Management: Directory

- Directory: mapping between file name and file descriptor
 - The OS could maintain a single directory for the whole disk
 - Problem: Need for unique name for files across all users
 - e.g., we could have only one file called program.c in the entire system

Name Management: Directory

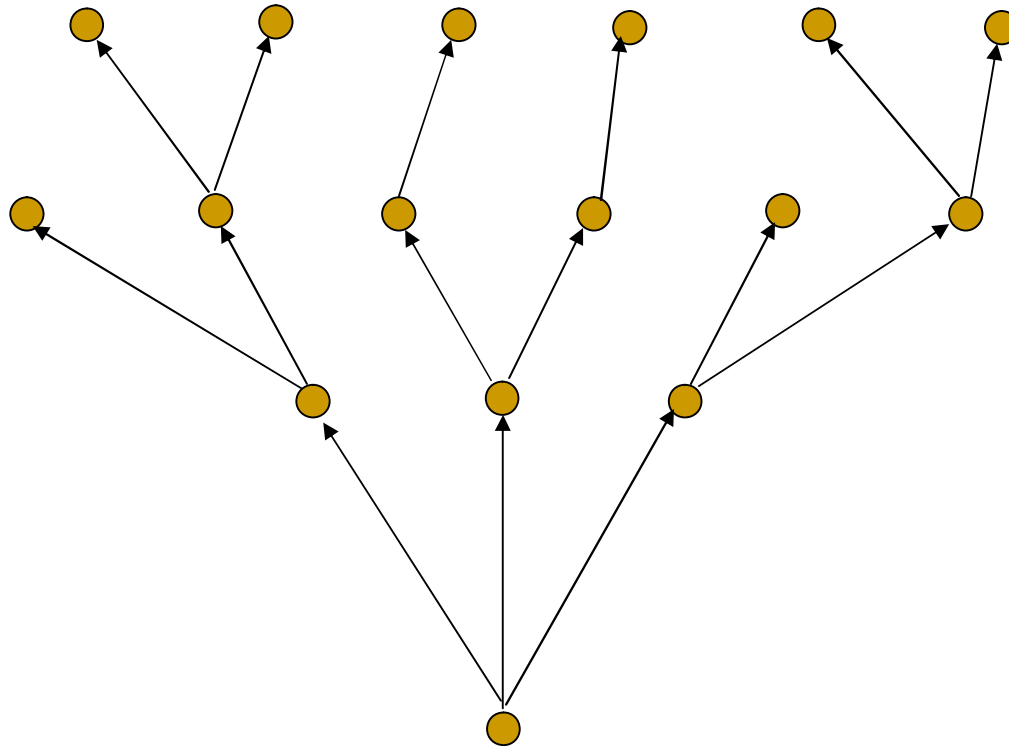
- Directory: mapping between file name and file descriptor
 - The OS could maintain a single directory for the whole disk
 - The OS could maintain a separate directory for each user
 - My directory would be referred to when I try to access a file
 - Then each user could have a file called program.c
 - But only one file called README

Name Management: Directory

- Directory: mapping between file name and file descriptor
 - The OS could maintain a single directory for the whole disk
 - The OS could maintain a separate directory for each user
 - The OS could maintain a single tree structured hierarchy of directories
 - e.g., UNIX, Linux

Tree Structured Directory Hierarchy

- Tree: A kind of data structure



Tree Structured Directory Hierarchy

■ Tree: A kind of data structure

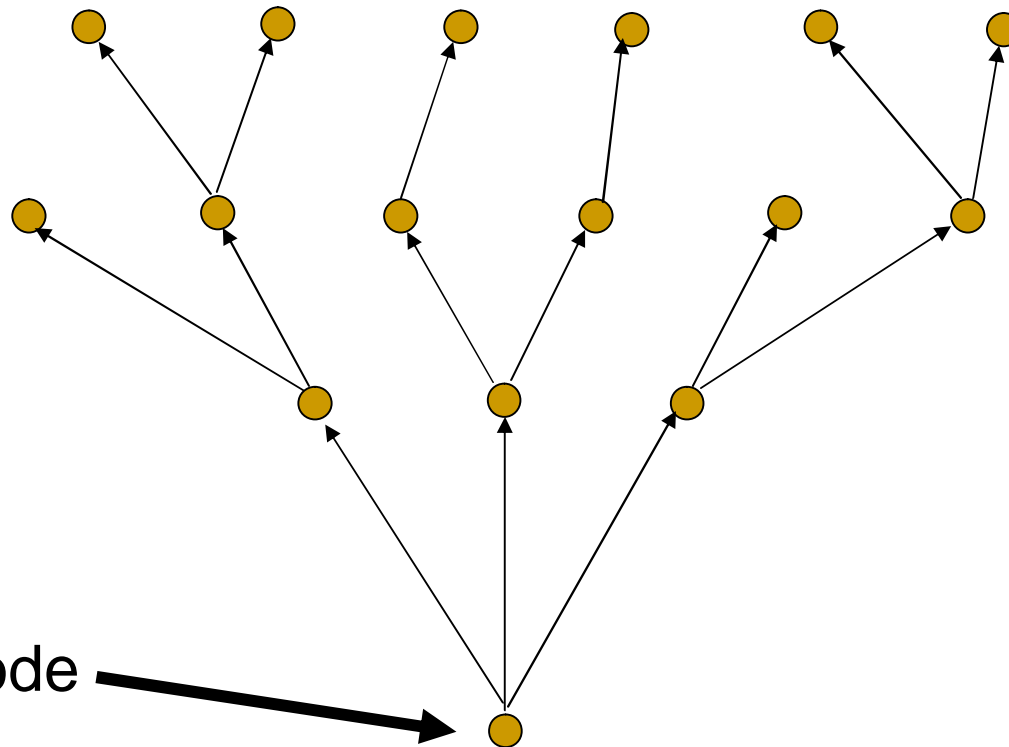
● node or vertex

→ edge

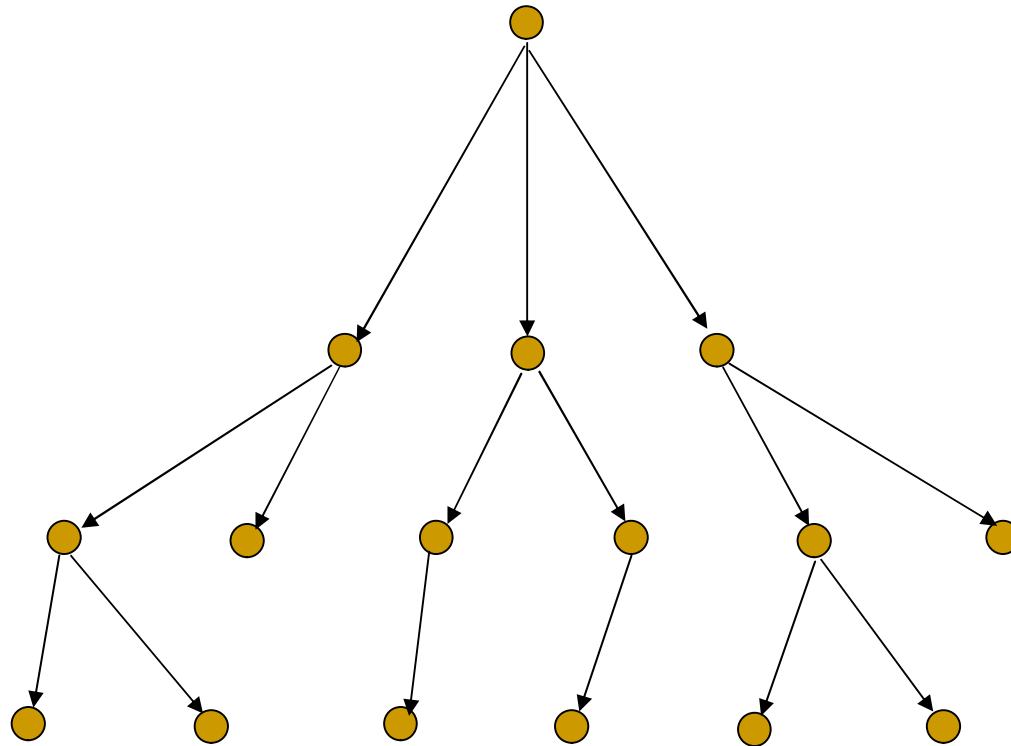
from parent node
to child node

leaf nodes or leaves

the root node



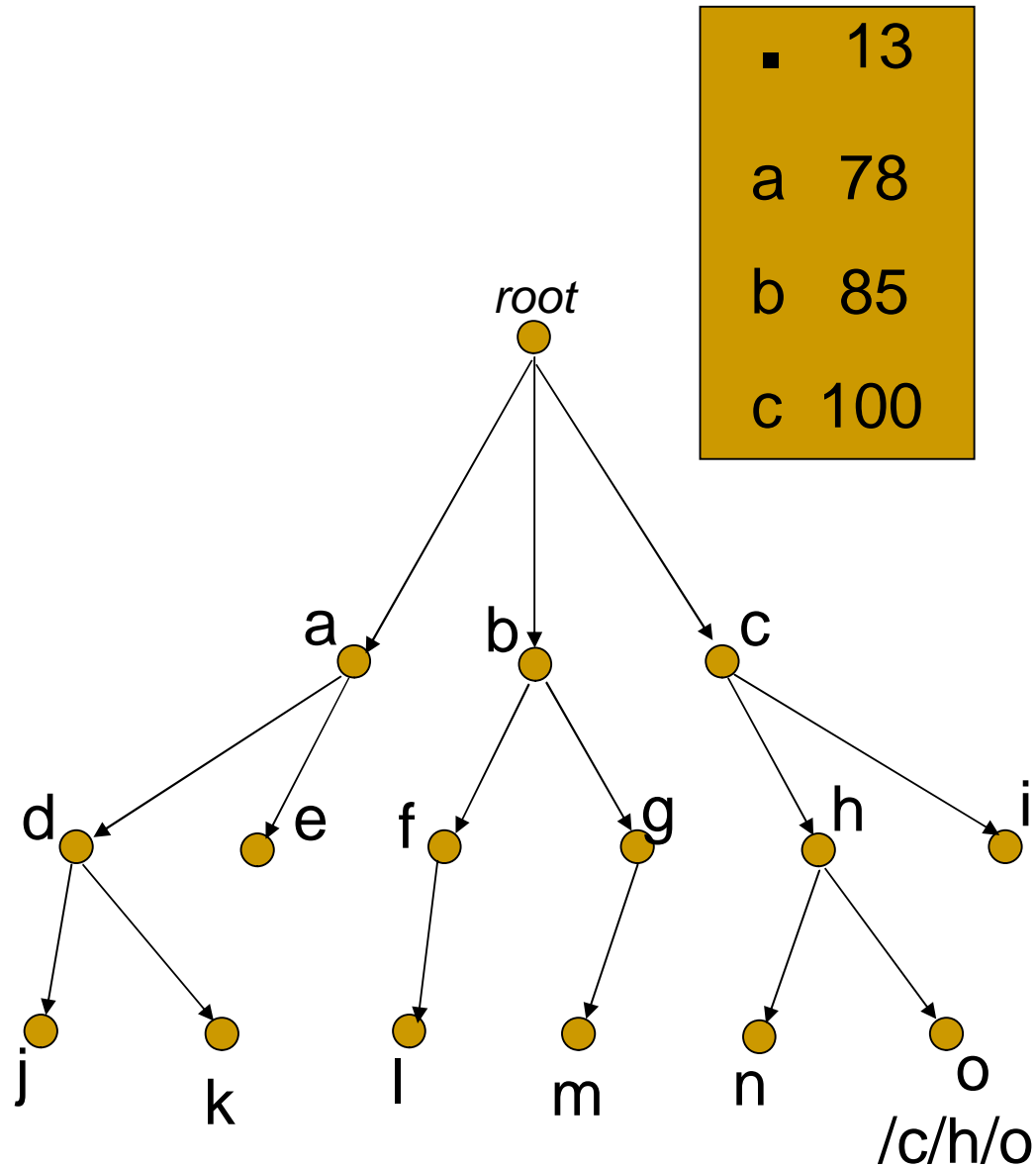
Tree Structured Directory Hierarchy



Tree Structured Directory Hierarchy

- Directories are stored on disk like regular files
- Each contains (filename, index node) pairs
- Each contains an entry with name `.` for itself
- Special (nameless) directory called the **root**
- Each file has a **pathname** that starts from the root directory
 - / separated sequence of directories

Tree Structured Directory Hierarchy



File System Design Issues

1. Disk management: efficient use of disk space
2. Name management: how users select files for use
3. **Protection**: of files from users

Protection

Objective: to prevent accidental or intentional misuse of a file system

- Aspects of a protection mechanism
 1. User identification (authentication)
 - Establishing that the user is who he/she claims to be
 - Could be done using userid/password, and password verification on login
 - Passwords must be stored securely
 - There are other possibilities
 - Biometrics (fingerprints, iris recognition, etc)

Protection

Objective: to prevent accidental or intentional misuse of a file system

- Aspects of a protection mechanism
 1. User identification (authentication)
 2. Authorization determination: determining what the user is entitled to do to the file
 - the system must keep track of what operations each user is allowed to do to each file
 - But that could be a huge amount of information

UNIX: 9 access permission bits

- Divides the universe of users into 3, in connection with a given file
 1. The Owner of the file
 2. The associates of the owner of the file
 - “users in the same group as the owner of the file”
 3. The other users
- Associates 3 permissions with each of these for that file
 1. Read permission
 2. Write permission
 3. Execute permission

Protection

Objective: to prevent accidental or intentional misuse of a file system

- Aspects of a protection mechanism
 1. User identification (authentication)
 2. Authorization determination: determining what the user is entitled to do to the file
 3. **Access enforcement**
 - preventing users from doing unauthorized access