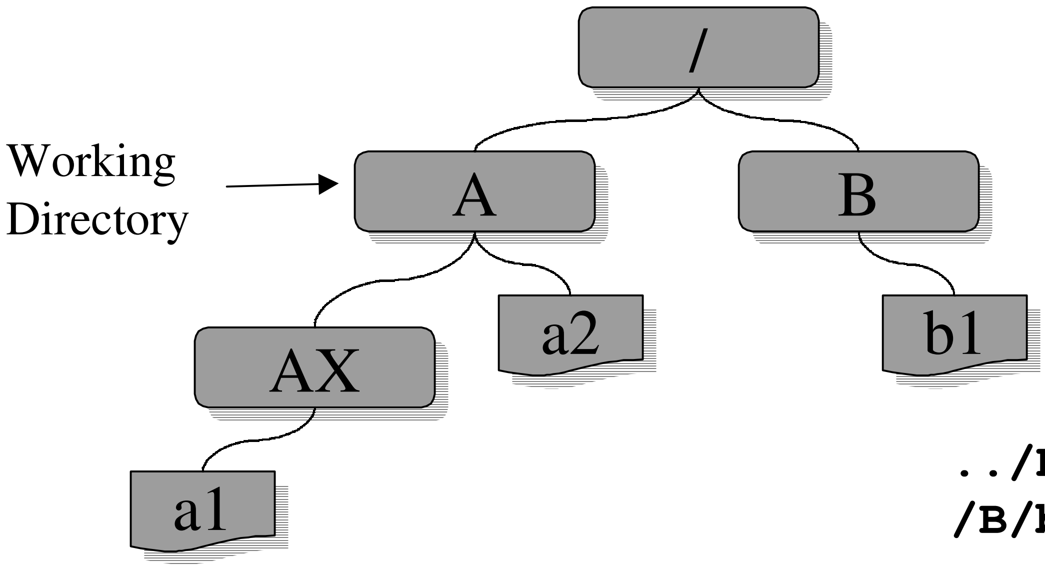
Alternative Schemes

- All files in *one* directory (!!)
- Each user has *one* directory
- Users can create directory
  - User controlled hierarchy

Example Directory Structure



`../B/b1` or
`/B/b1`

`AX/a1` or
`/A/AX/a1`

Notes

- Directories contain:
  - file names
  - directory names

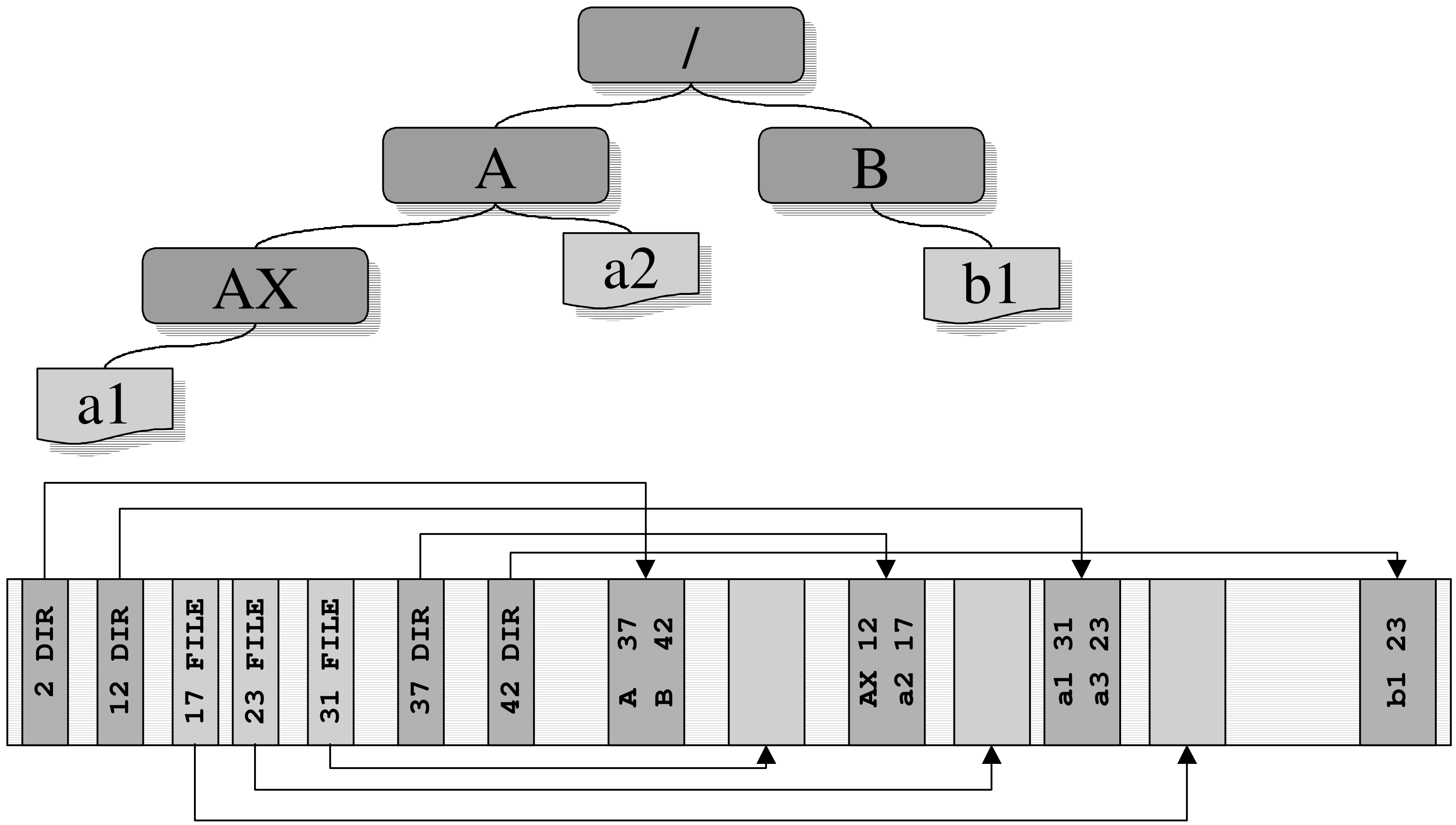
- Each process has:
  - working directory
  - inherited on creation
  - filenames relative to current working directory

Implementing Naming

- Directories map:
  - names $\rightarrow$ System File ID (SFID)
  - (and vice-versa?)

- Directory contains:
  - list of (name, SFID) pairs

Implementing Naming



File Sharing

- Any application can generate names independent of username
  - /etc/password
  - /lib/libc.a
  - /boot/vmlinuz-2.2.1
- Protection must be applied independently of naming (how?)

Files with Multiple Names

- Hard Link (ln exists new)
  - exists must be present
  - new becomes first-class name for same file
- Soft Link (ln -s anyname new)
  - anyname may or may not exist
  - new is translated (at open() time) into anyname
- Windows® shortcuts?

Existence Control

- File may have multiple names:
  - /etc/sendmail
  - /usr/bin/mailq
  - /root/bin/newaliases
- Any name may be deleted from directory

- When should file storage space be released?