

File and Directories

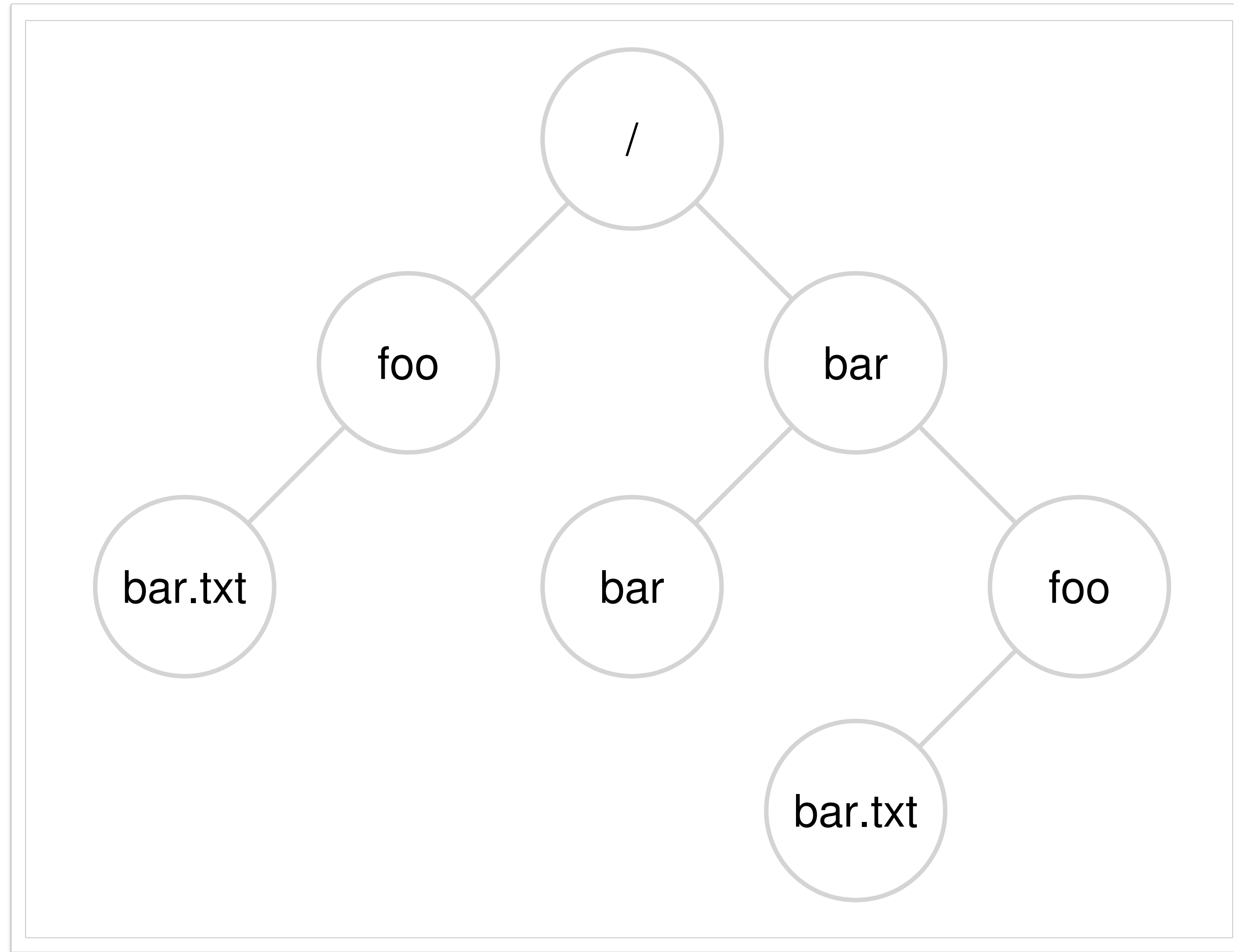
Virtualization of storage

Two key abstractions

» File

» Directory

Directory hierarchy



File system interface: Creating files

```
int fd = open("foo", O_CREAT | O_WRONLY | O_TRUNC);
```

File system interface: Reading and writing files

```
prompt> echo hello > foo  
prompt> cat foo  
hello  
prompt>
```

File system interface: Reading and writing files

```
prompt> strace cat foo
...
open("foo", O_RDONLY|O_LARGEFILE)      = 3
read(3, "hello\n", 4096)                = 6
write(1, "hello\n", 6)                  = 6
hello
read(3, "", 4096)                        = 0
close(3)                                  = 0
...
prompt>
```

File system interface: Writing immediately with `fsync()`

`write()`: Please, write this data to the disk at some point in the future. The data is written to a buffer, and later written to the persistent storage.

Data loss is rare. But it could happen if program crashes before the data is actually transferred to the disk.

File system interface: Writing
immediately with `fsync()`

Application such as Data Base Management
Systems need sometimes to force writes to disk.

```
fsync(int fd);
```


File system interface: Writing immediately with `fsync()`

```
int fd = open("foo", O_CREAT | O_WRONLY | O_TRUNC);  
assert(fd > -1);  
int rc = write(fd, buffer, size);  
assert(rc == size);  
rc = fsync(fd);  
assert(rc == 0);
```

File system interface: Renaming files

```
mv foo bar
```

`strace` shows that `mv` calls the system call `rename()`.

File system interface: Renaming files

- `mv foo bar`
- `strace` shows that `mv` calls the system call `rename(char *old, char *new)`
- `rename()` is implemented as an atomic call.

File system interface: Renaming files

- **Example:** Editing a file using a text editor (e.g., emacs). Editor needs to guarantee that the new file has the original content as well as the recently updated text.

File system interface: Renaming files

```
int fd = open("foo.txt.tmp", O_WRONLY|O_CREAT|O_TRUNC);  
write(fd, buffer, size); // write out new version of file  
fsync(fd);  
close(fd);  
rename("foo.txt.tmp", "foo.txt");
```

File system interface: Removing files

- `rm foo`

File system interface: Removing files

- `rm foo`

```
prompt> strace rm foo
```

```
...
```

```
unlink("foo") = 0
```

```
...
```

File system interface: Creating directories

```
prompt> strace mkdir foo
...
mkdir("foo", 0777) = 0
...
prompt>
```


File system interface: Creating directories

```
prompt> strace mkdir foo
...
mkdir("foo", 0777) = 0
...
prompt>
```

```
prompt> ls -a
./  ../
```