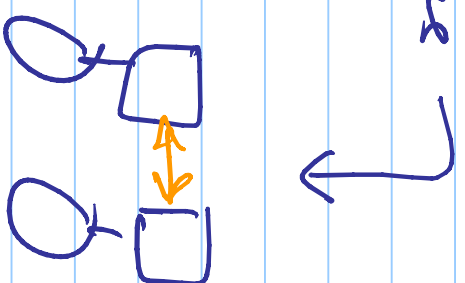


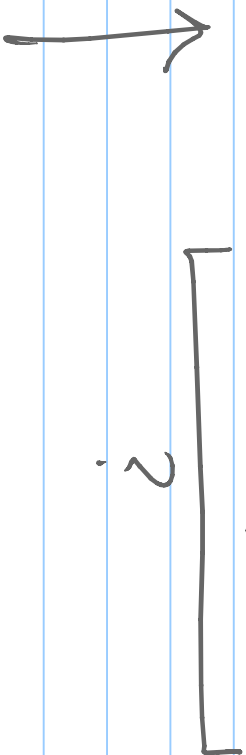
History

- timesharing → 1 computer
 - many interactive users
- collaboration
 - single file systems
 - all users are visible

- Personal Computer - stand-alone
- " "
- Network file systems
- Database Workstation



Network File system



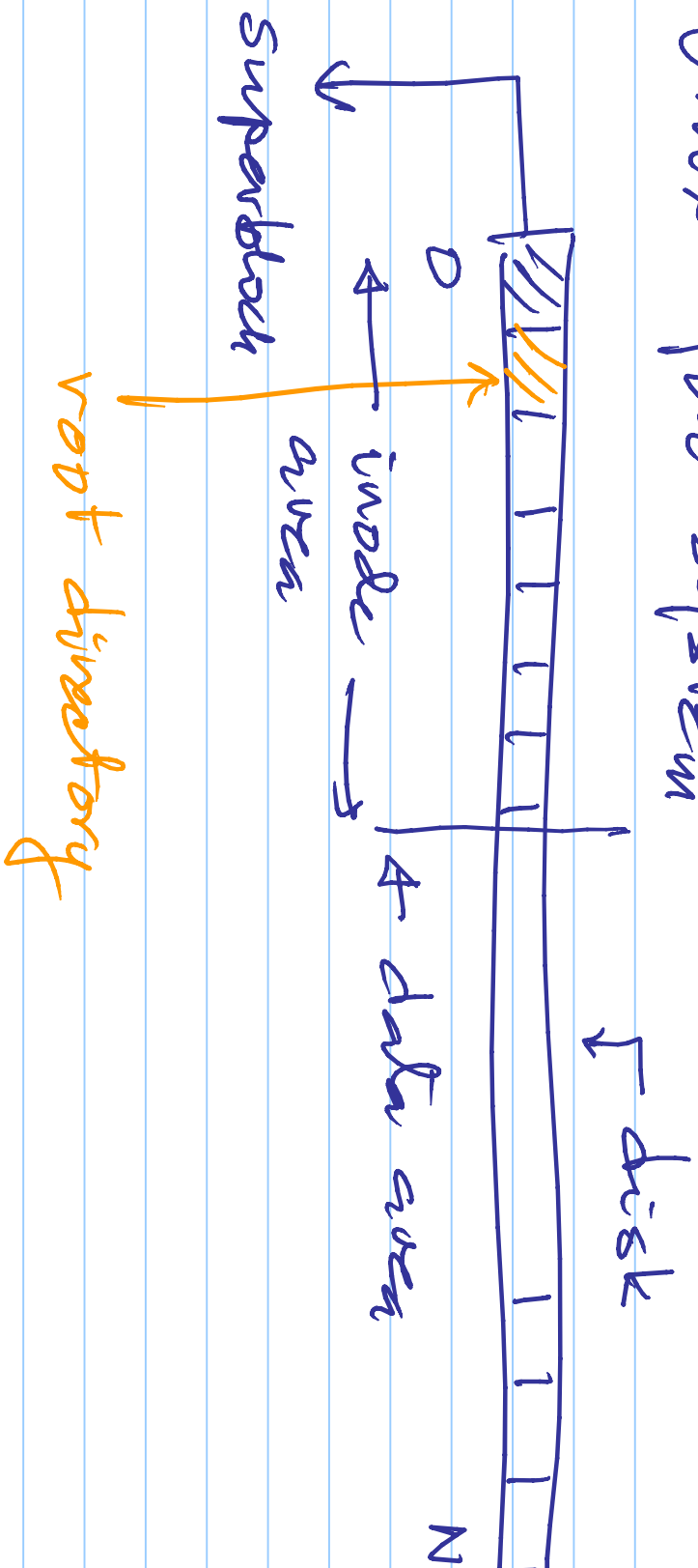
- makes a set of file systems appear as one file system
- location of the files are transparent
- backward compatible

File system $\rightarrow$. directories & files]
• tree structure

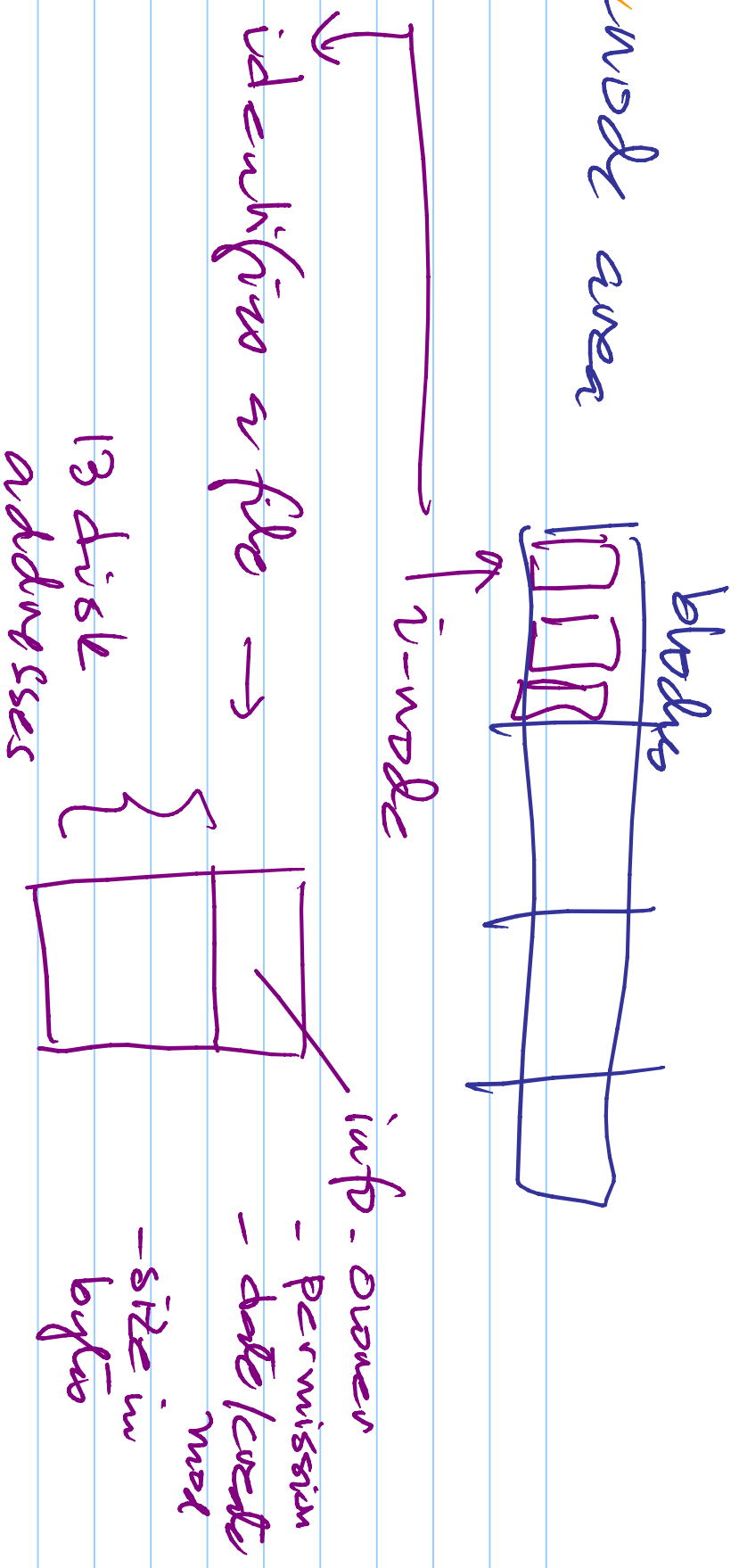
Disk $\rightarrow$ linear sequence of
storage blocks.

Disk + on-disk-data structure &
(metadata)

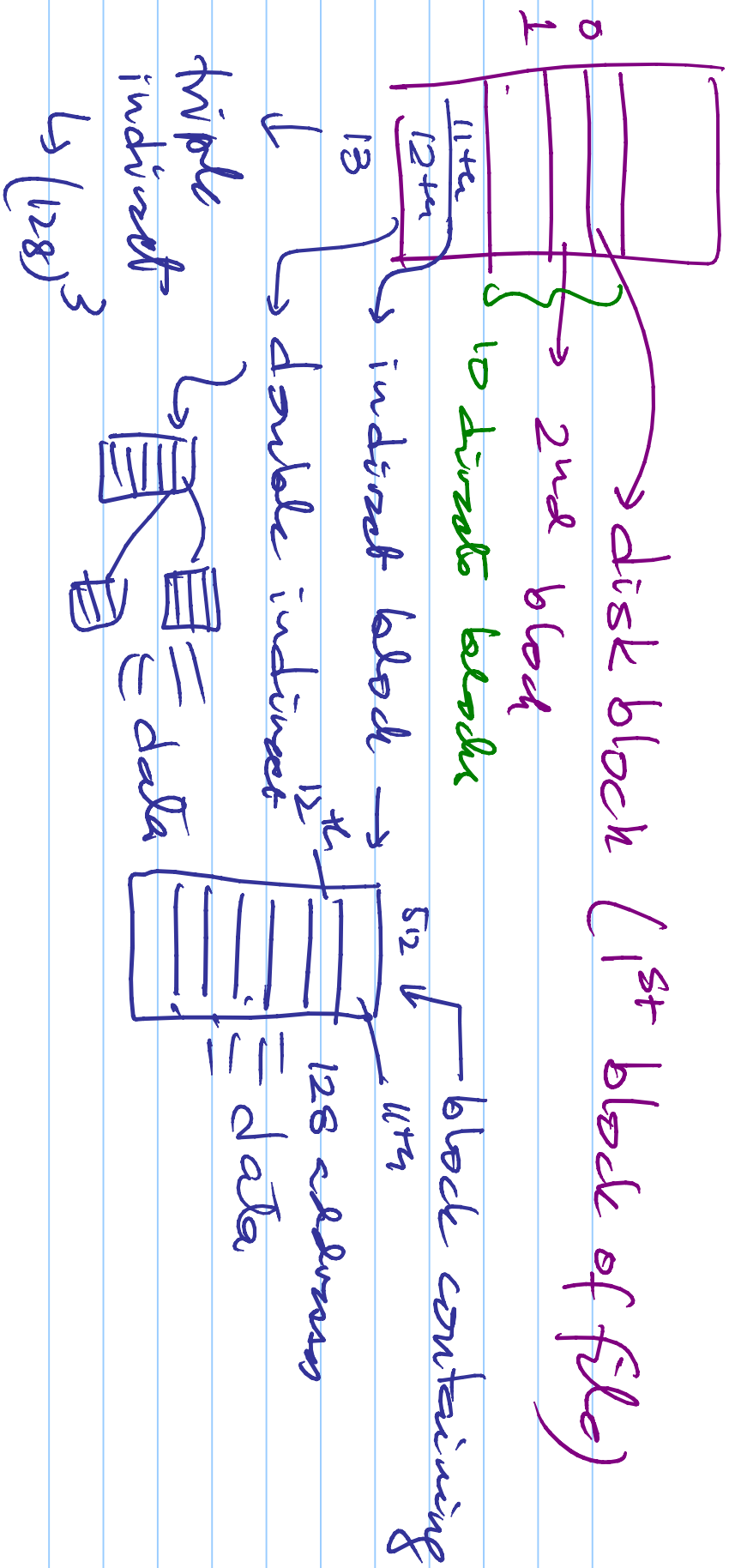
Unix file system



inode area

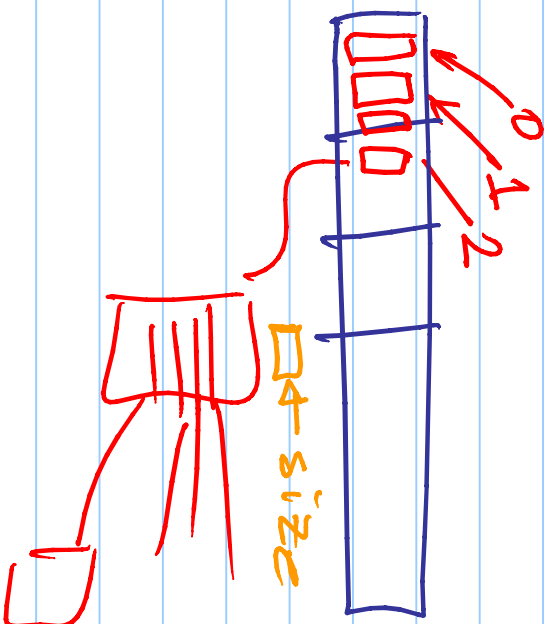


13



Every file has a i-number

↳ address of i-node



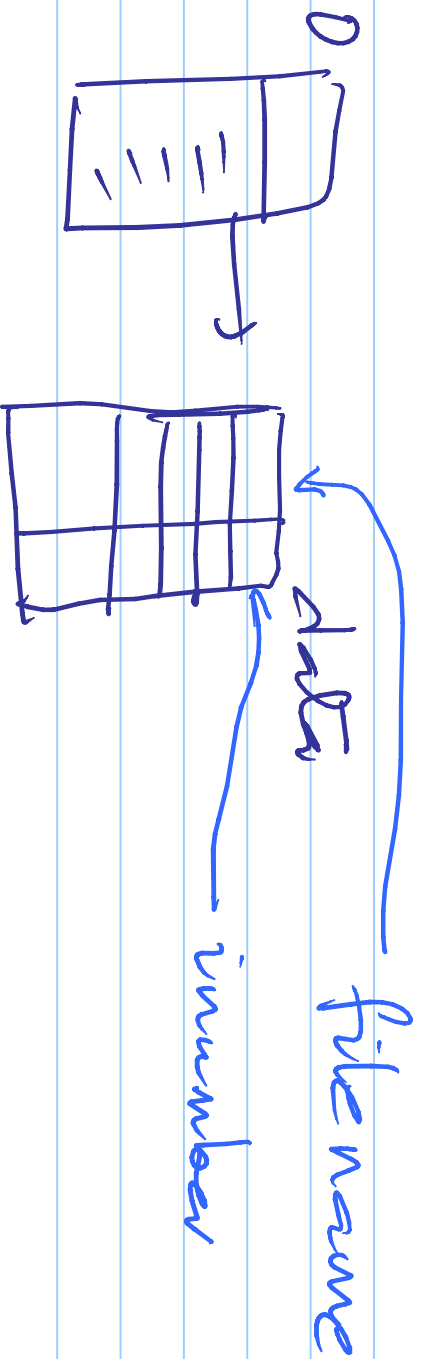
Read/write

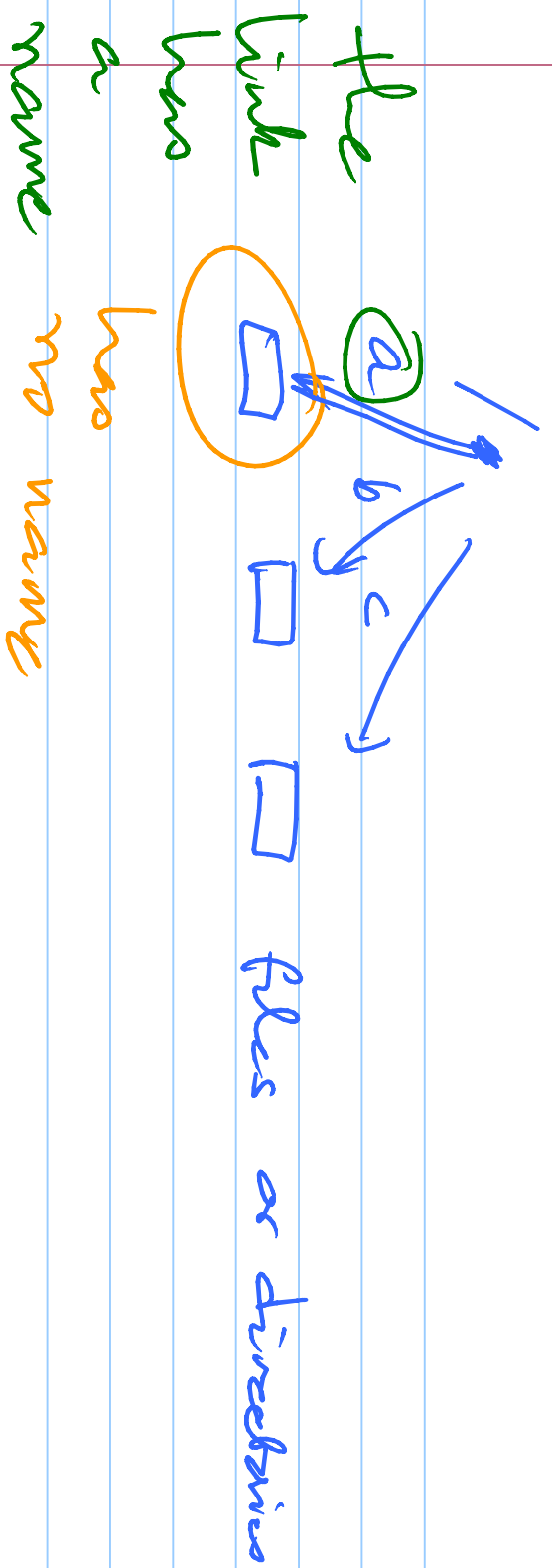
- needs i-number
& address to the
inode

directory

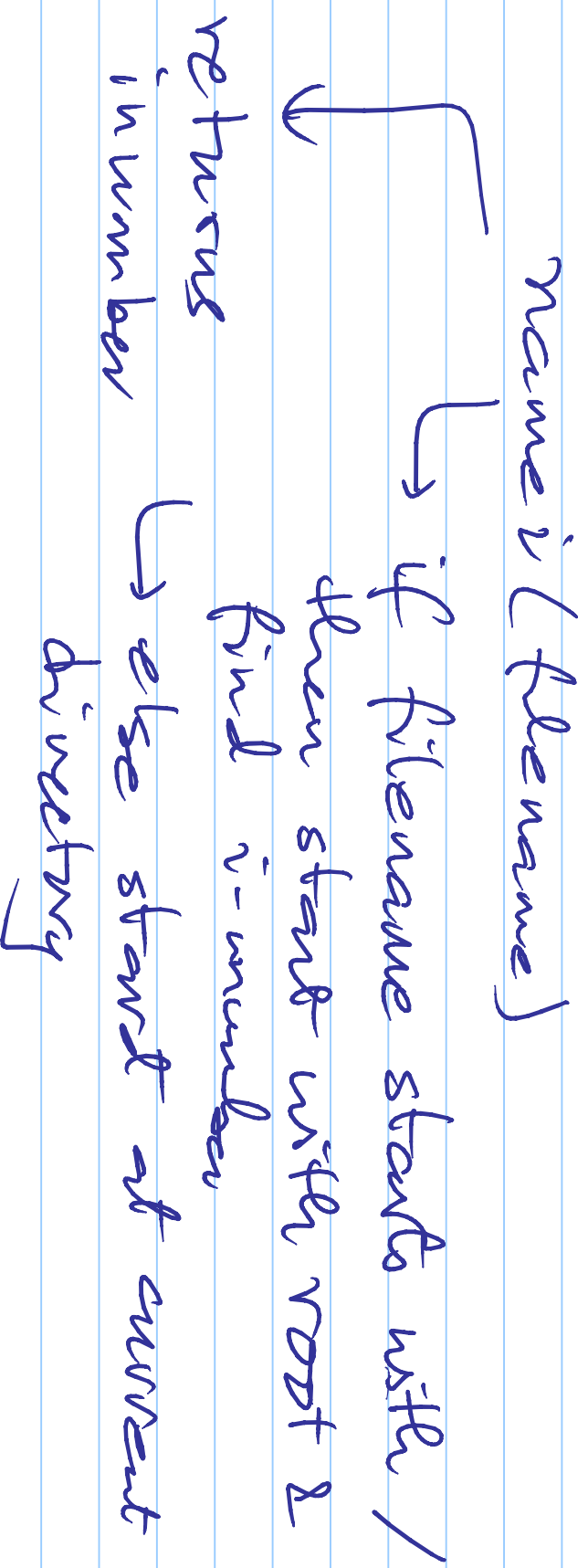
① Nameless directory / (root)

↳ has i-number = 0





Kernel operations



Open ( filename
system call

```
get number (name i)
```

```

    put number in a table
    return index to table }
    _____

```

flu descriptors

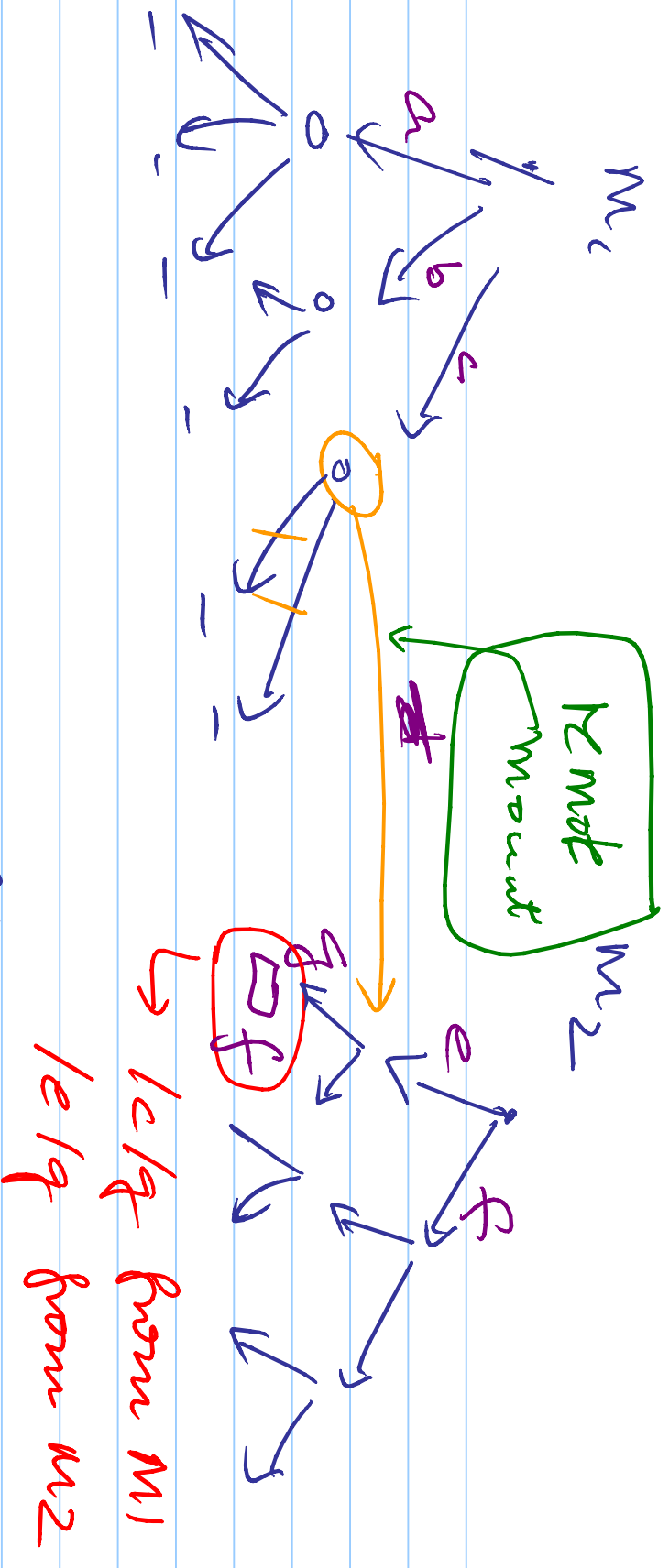
read (fd, ...)

{ from fd get number

from file pointer get block location

read data from file...

}

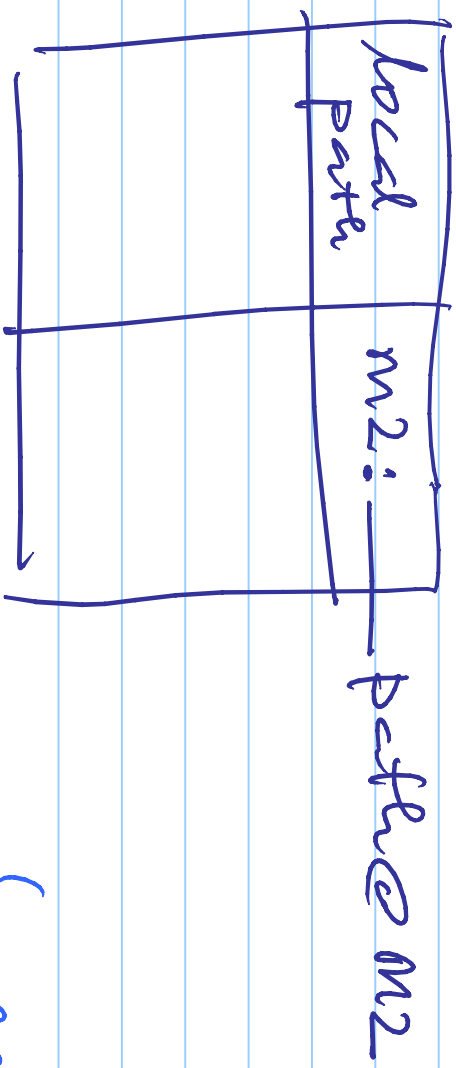


NFS → network file system

remote mount

↳ needs a mount table @ M1

M1's
mount
table



→ patch name i , read & use nsfd (prev)

Name i (path)

↳ every time see a dir → check mount table

→ if mount table entry found

→ rewrite local path to remote path

→ send request to nsfd →
↳ will do a local name i → & get i-number

Read() → if local →

else

Send i number & other info
to remote msfd

get data

return to app

NFS (originated from Sun)

→ used a "stateless" file server

→ the server has no memory
→ it forgets prior requests from client

Stateless file service ~~problems~~

- ① file pointer not in server
- ② file server does not know file is open $\rightarrow$ may delete it
- ③ But file server failure $\rightarrow$ does not matter