# CS341 #27 — Files and Filesystem

*TCP Server hints*
getaddrinfo(**NULL**, **"0"**, &hints, &result); // ANY Port
**sock_fd** = socket( ....)
// Later, after calling bind ? *Spot the mistake ....* (what's missing?)
struct sockaddr_in sin;
socklen_t socklen = sizeof(sin);
if (getsockname(sock_fd, (struct sockaddr *)&sin, &socklen) == 0) printf("port %d\n", sin.sin_port);
ntohs( )
See also : Reusing ports.

Design a file system! What are your design goals?

hierarchies + searching indexing
read/write
organization
file types
access control
encryption
compression
Robust! Tolerant HW errors
Security!

parent director

Paths "." ".." "..." ? "foo1/foo2" vs "/bar1/bar2"?
relative path absolute path
Example of relative path?

Can you simplify a/b/../c/. ? ⟺ a/c

What is an absolute path? /home/Sam/a/c

What if I need the **Absolute Canonical** Path?
only one way
TREE?

```
int main() {
  char* path = realpath("./../../",NULL);
  puts(path);
  free(path);
  return 0;
}
```

Casestudy: Use realpath to secure file access of a webserver?

```
  char* canonbasepath = realpath(basedir,NULL);
  asprintf( &file, "%s/%s", canonbasepath , url );
  puts(file)
```

File systems are block-based. Why make disk blocks the same size as memory pages?

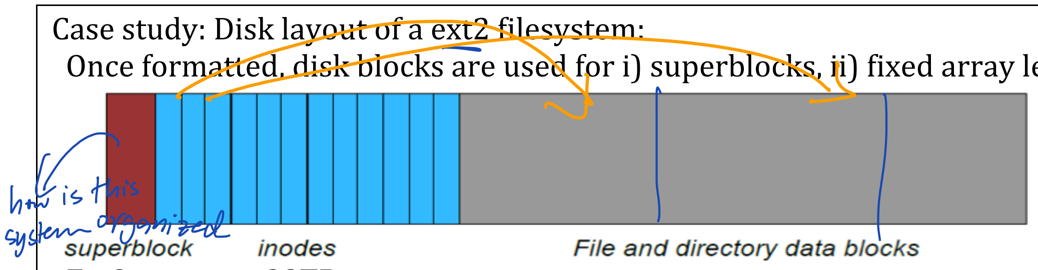
What do we want to store for each file?
mime type
permission
owner
name is not a part of file?
data
time
date
size
name

What is an inode? Which of the above items is stored in the inode?

Connection between inode & stat?

Case study: Disk layout of a ext2 filesystem:

Once formatted, disk blocks are used for i) superblocks, ii) fixed array length of inodes and iii) data blocks:



Ext2 supports 32TB storage.

Ext3 (2001): supports journaling.

Ext4 (2008): Performance + (16TB files) upto 1EB (1024 Peta B) storage.

How does an inode index the file contents?

4096 B / 4B = 1024 entries

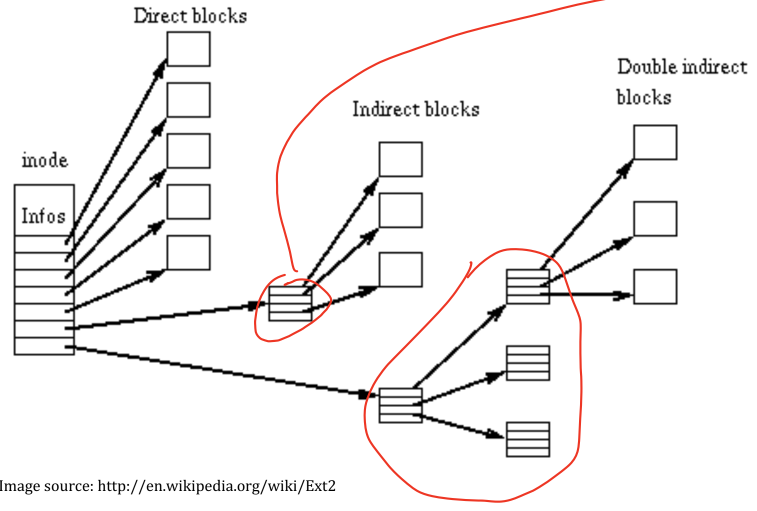


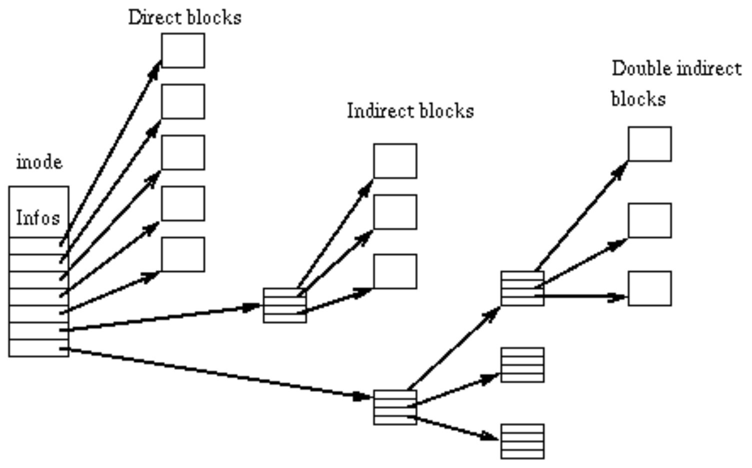
Image source: http://en.wikipedia.org/wiki/Ext2

In the following examples assume an ext2 filesystem with 4KB disk blocks. Files use 10 direct blocks and 2^32 addressable disk blocks.

1. How many indirect blocks can be referenced?

2. How large is the file (in blocks of data) if the indirect block index is half full?

3. What is the total number of blocks used (ignore the inode entry in the inode array)?



What about huge files? Do we need triple indirect? Quad indirect?