Let's pretend we’re a filesystem…

One cat per bucket, please.
Let’s pretend we’re a filesystem…
The UNIX Filesystem

- a disk can be divided into logical *partitions*, as described in e.g. the MBR
The UNIX Filesystem

- a disk can be divided into logical partitions, as described in e.g. the MBR
- a logical partition has a disklabel, describing the geometry of the disk and the filesystem partitions
The UNIX Filesystem

- A disk can be divided into logical partitions, as described in e.g. the MBR.
- A logical partition has a disklabel, describing the geometry of the disk and the filesystem partitions.
- A filesystem partition is a collection of cylinder groups, on which you can create a new filesystem.

The code snippet shown is an example of viewing a disk label using the `disklabel` command in a terminal:

```bash
ip-10-10-0-47# disklabel -C xbd0
#/dev/rxbd0:
type: ESDI
disk: image
label:
flags:
bytes/sector: 512
sectors/track: 32
tracks/cylinder: 64
sectors/cylinder: 2048
cylinders: 1907
total sectors: 20971520
rpm: 3600
interleave: 1
trackskew: 0
cylinderskew: 0
headswitch: 0
headskew: 0
track-to-track seek: 0
# microseconds
8 partitions: # size offset ftype [fs size bsize cpg/sgs]
a: 10239/0/0 1/0/0 4.28SD 1024 8192 16 # (Cyl. 1 - 10239)
c: 1906/0/0 1/0/0 unused 0 0 # (Cyl. 1 - 1906)
d: 10240/0/0 0/0/0 unused 0 0 # (Cyl. 0 - 10239)
```
- each cylinder group contains a list of inodes as well as the actual directory- and data blocks
The UNIX Filesystem

• divided into cylinder groups
• filesystem metadata replicated in superblocks in each cylinder group
• filesystem block size can be different from physical block size
• inodes separated from data blocks
• inode density is fixed at filesystem creation time
• various options allow tuning of filesystem to accommodate anticipated use

Next time: file types and mount points
Links

File Systems and Storage Models:

A Fast File System for UNIX:
https://people.eecs.berkeley.edu/~brewer/cs262/FFS.pdf

CS631 Video on the Unix Filesystem:
https://youtu.be/kY4JAXYyByQ

Manual pages:
fs(5), dumps(8), newfs(8)