Filesystems

Requirements

There are a lot of things we want from files

Filesystems

Requirements

There are a lot of things we want from files

- create a new file

Filesystems

Requirements

There are a lot of things we want from files

- create a new file
- delete a file

Filesystems

Requirements

There are a lot of things we want from files

- create a new file
- delete a file
- open a file to access it

Filesystems

Requirements

There are a lot of things we want from files

- create a new file
- delete a file
- open a file to access it
- read data from a file

Filesystems

Requirements

There are a lot of things we want from files

- create a new file
- delete a file
- open a file to access it
- read data from a file
- write data to a file

Filesystems

Requirements

There are a lot of things we want from files

- create a new file
- delete a file
- open a file to access it
- read data from a file
- write data to a file
- close a file when we are done

Filesystems

Requirements

There are a lot of things we want from files

- create a new file
- delete a file
- open a file to access it
- read data from a file
- write data to a file
- close a file when we are done
- rename a file

Filesystems

Requirements

There are a lot of things we want from files

- create a new file
- delete a file
- open a file to access it
- read data from a file
- write data to a file
- close a file when we are done
- (rename a file)

That last one is actually a directory operation as we shall see in a moment

Filesystems

Requirements

And directories

Filesystems

Requirements

And directories

- create a new directory

Filesystems

Requirements

And directories

- create a new directory
- delete a directory

Filesystems

Requirements

And directories

- create a new directory
- delete a directory
- scan a directory for a filename or directory name

Filesystems

Requirements

And directories

- create a new directory
- delete a directory
- scan a directory for a filename or directory name
- add a file to a directory

Filesystems

Requirements

And directories

- create a new directory
- delete a directory
- scan a directory for a filename or directory name
- add a file to a directory
- remove a file from a directory

Filesystems

Requirements

And directories

- create a new directory
- delete a directory
- scan a directory for a filename or directory name
- add a file to a directory
- remove a file from a directory
- rename a file

Filesystems

Requirements

And directories

- create a new directory
- delete a directory
- scan a directory for a filename or directory name
- add a file to a directory
- remove a file from a directory
- rename a file

The last three are intertwined

Filesystems

Requirements

This all is before we come to things like

Filesystems

Requirements

This all is before we come to things like

- speed of access

Filesystems

Requirements

This all is before we come to things like

- speed of access
- speed of update

Filesystems

Requirements

This all is before we come to things like

- speed of access
- speed of update
- scalability to large numbers of files

Filesystems

Requirements

This all is before we come to things like

- speed of access
- speed of update
- scalability to large numbers of files
- efficient use of disk space

Filesystems

Requirements

This all is before we come to things like

- speed of access
- speed of update
- scalability to large numbers of files
- efficient use of disk space
- reliability

Filesystems

Requirements

This all is before we come to things like

- speed of access
- speed of update
- scalability to large numbers of files
- efficient use of disk space
- reliability
- protection/security

Filesystems

Requirements

This all is before we come to things like

- speed of access
- speed of update
- scalability to large numbers of files
- efficient use of disk space
- reliability
- protection/security
- simple backup and recovery

Filesystems

We shall be looking at the classical Unix filesystem as an example

Filesystems

We shall be looking at the classical Unix filesystem as an example

Other filesystems are similar in their principles, though modern filesystems are immensely tweaked and tuned

Filesystems

We shall be looking at the classical Unix filesystem as an example

Other filesystems are similar in their principles, though modern filesystems are immensely tweaked and tuned

They vary in their choice of datastructures and algorithms to implement the hierarchy for efficiency or other reasons

Filesystems

Records

Modern files tend to be essentially long arrays of bytes with no further structure

Filesystems

Records

Modern files tend to be essentially long arrays of bytes with no further structure

Early files had structure, namely *records*

Filesystems

Records

Modern files tend to be essentially long arrays of bytes with no further structure

Early files had structure, namely *records*

This was a hangover from early systems using things like punched cards

Filesystems

Records

Modern files tend to be essentially long arrays of bytes with no further structure

Early files had structure, namely *records*

This was a hangover from early systems using things like punched cards

A record is a fixed-size block of data, say 80 bytes

Filesystems

Records

Modern files tend to be essentially long arrays of bytes with no further structure

Early files had structure, namely *records*

This was a hangover from early systems using things like punched cards

A record is a fixed-size block of data, say 80 bytes

Records could only be read or written as a whole: this meant implementation on the hardware of the time was easy

Filesystems

Records

It also aligned with the way data was regarded at the time: records of peoples names, job classification, salary and so on (*fields*)

□

Filesystems

Records

It also aligned with the way data was regarded at the time: records of peoples names, job classification, salary and so on (*fields*)

They would expect an entire record to be read or written at once

□

Filesystems

Records

It also aligned with the way data was regarded at the time: records of peoples names, job classification, salary and so on (*fields*)

They would expect an entire record to be read or written at once

Modern filesystems are *byte oriented* and you can access them however you please

□