

Assignment Project Exam Help

(w <https://eduassistpro.github.io>
axgopala@imperial.ac.uk

Add WeChat edu_assist_pr

Long term non-volatile, online storage → e.g. programs, data, text, photos, music, ...

Shar

appli

Conc

<https://eduassistpro.github.io>, ...

Organisation and management of data

directories, symbolic names, backups, snapshots

File: Named collection of data of arbitrary size

File Naming

Typical file extensions

File Type	Usual Extension	Function
executable	exe, com, bin	read to run machine-language pro-
obj		not
source code	c, cc, java, py, hs	sour
batch	bat, sh	com prete
...
...

What is a file?

Assignment Project Exam Help

<https://eduassistpro.github.io>

Add WeChat edu_assist_pr

Create	Create empty file. Allocate space and add to directory
Delete	Deallocate space. Invalidate or remove directory entry
Open	Search directory for file name. Check access validity and set pointers to file
Close	
Read	
Write	Access file, update pointers
Reposition/seek	Set current position in file to
Truncate	Erase contents but keep all
Rename	Change file name
Read attributes	e.g. creation date, size, archive flag, ...
Write attributes	e.g. protection, immutable flag, ...

System Call	Description
<code>open (file, how, ...)</code>	Open a file for reading/writing
<code>close (fd)</code>	Close file
<code>read (fd, buf, nbytes)</code>	Read
<code>write (fd, buf, nbytes)</code>	Write
<code>lseek (fd, offset, ...)</code>	Move
<code>stat (name, &buf)</code>	Get file
<code>fcntl (fd, cmd, ...)</code>	File locking and other operations

Logical name to physical disk address translation

- i.e. `/homes/axgopala/vimrc` → disk 2, block 399

Management of disk space

-

File I/O

Perf

- Caching and buffering

Protection against system failure

- Back-up and restore

Security

- Protection against unauthorised access

File Attributes I

Basic information

file name	symbolic name; unique within directory
file type	text, binary, executable, directory, ...
file o	
file c	

Address information

volume	disk drive, partition
start addresses	cyl, head, sect, LBA
size used	
size allocated	

Access control information

owner authentication per	person who controls file (often creator) password
--------------------------------	--

Usage

creation timestamp last modified last read access activity counts	date and time
--	---------------

Assignment Project Exam Help

<https://eduassistpro.github.io>

Add WeChat edu_assist_pr

Assignment Project Exam Help

<https://eduassistpro.github.io>

Add WeChat edu_assist_pr

Assignment Project Exam Help

<https://eduassistpro.github.io>

Add WeChat edu_assist_pr

Assignment Project Exam Help

<https://eduassistpro.github.io>

Add WeChat edu_assist_pr

Assignment Project Exam Help

<https://eduassistpro.github.io>

Add WeChat edu_assist_pr

Assignment Project Exam Help

<https://eduassistpro.github.io>

Add WeChat edu_assist_pr

Assignment Project Exam Help

<https://eduassistpro.github.io>

Add WeChat edu_assist_pr

Assignment Project Exam Help

<https://eduassistpro.github.io>

Add WeChat edu_assist_pr

Unix/Linux stat System Call

File attributes can be accessed using system call `stat(2)` (man 2 `stat`)

- Return information about specified file in struct `stat`

```
struct stat {  
    de          st          /* ID o  
    in          st          /* in  
    mo  
    nl  
    uid_t      st_uid;      /* user ID of owner */  
    gid_t      st_gid;      /* group ID of owner */  
    ...  
    off_t      st_size;      /* total size, in bytes */  
    struct timespec st_atim; /* time of last access */  
    struct timespec st_mtim; /* time of last modification */  
    struct timespec st_ctim; /* time of last status change */  
};
```

File size naturally variable

Space allocated in blocks (typically 512 – 8192 bytes)

Choosing block size

- <https://eduassistpro.github.io>
 - More memory needed for buffer space
- Block size too small → wastes space
 - High overhead in terms of management
 - High file transfer time: seek time greater than transfer time

Which allocation works the best?

Assignment Project Exam Help

Contiguous file allocation

Block

<https://eduassistpro.github.io>

File allocation

Index blocks

Add WeChat edu_assist_pro

Place file data at contiguous addresses on storage device

Advantages

- Successive logical records typically physically adjacent

Disadvantages

- <https://eduassistpro.github.io>
- Poor performance if files grow and shrink over time
- File grows beyond size originally specified and must be moved to new area of adequate size if no free blocks available
 - Must be transferred to new area of adequate size
 - Leads to additional I/O operations

Assignment Project Exam Help

<https://eduassistpro.github.io>

Add WeChat edu_assist_pr

- (a) Contiguous allocation of disk space for seven files
- (b) The state of the disk after files D and F have been removed

Block Linkage (Chaining) I

Place file data by linking them together \Rightarrow insertion/deletion by modifying pointer in previous block

Assignment Project Exam Help

<https://eduassistpro.github.io>

Add WeChat edu_assist_pr

Assignment Project Exam Help

Need to search list to find data block



- <https://eduassistpro.github.io>

- Block-to-block seeks occur

Wastes pointer space in each block

Add WeChat edu_assist_pr

Block Allocation Table I

Store pointers to file blocks

- Directory entries indicate first block of file
- Block number as index into block allocation table
 - Determines location of next block
-

File Address
to Block

- Stored on disk but cached in memory for performance

Reduces number of lengthy seeks to access given file

- But files become fragmented → periodic defragmentation
- Table can get very large

Assignment Project Exam Help

<https://eduassistpro.github.io>

Add WeChat edu_assist_pr

Block Linkage vs. FAT

Consider a disk with a block size of 1024 bytes. Each disk address can be seen as a sequence of 10 bits. Each block contains 1024 bytes, each byte is 8 bits.

1

data byte and the 510100th data byte?

Hint: $500 \times 1020 = 510000$ and $498 \times 1024 = 509$

2

How does this change if a file allocation table (FAT) is used?

Example Problem

Answer: Block Linkage

There are 1020 data bytes per block. The 1022^{nd} byte is resident on the 2^{nd} disk block \rightarrow 2 reads are required

<https://eduassistpro.github.io>

The 510100^{th} data byte is resident in the 501
reads are required

01

Add WeChat edu_assist_pr

Example Problem

Answer: FAT

There are 1024 data bytes per block. Each block of the FAT can represent $\frac{1024}{256} = 4$ data blocks.

① The 1020th byte is on the 1st block and requires 1 read for the

②

<https://eduassistpro.github.io>

- At best, all of the first 499 blocks of the file can be represented in 2 FAT blocks

- At worst, 499 reads could be performed for

Either case requires 1 extra read for the data. H

- Best case requires 3 reads
- Worst case requires 500 reads

Answer: FAT

Assignment Project Exam Help

<https://eduassistpro.github.io>

Add WeChat edu_assist_pr

Answer: FAT

Assignment Project Exam Help

<https://eduassistpro.github.io>

Add WeChat edu_assist_pr

How can we improve?

Each file has one (or more) index blocks

- Contain list of pointers that point to file data blocks
- File's directory entry points to its index block



<https://eduassistpro.github.io>

Adv

- Searching may take place in index blocks the
- Place index blocks near corresponding data for fast access to data
- Can cache index blocks in memory for faster access

Assignment Project Exam Help

<https://eduassistpro.github.io>

Add WeChat edu_assist_pr

inode

Index blocks called **inodes** (index nodes) in UNIX/Linux

On file open, OS opens **inode table** →
inode entry created in memory

Str
clu

- 1 Disk device number
- 2 Inode number (for re-write)
- 3 Num of processes with opened file
- 4 Major/minor device number

Type and access control

Number of links

User ID

Group ID

Direct pointer

Indirect pointer

Double indirect pointer

Triple indirect pointer

Assignment Project Exam Help

<https://eduassistpro.github.io>

Add WeChat edu_assist_pr

Inodes 1

In a particular OS, an inode contains 6 direct pointers, 1 pointer to a (sin

Each

byte

① What is the maximum file size for this file system?

② What is the maximum file size if the OS would use indirect pointers?

Answer: Inodes I

1 The maximum file size is:

$$\begin{aligned} & 6 \times 1024 && \text{(data directly indexed)} \\ & + 128 \times 1024 && \text{(data referenced by single indirect)} \end{aligned}$$

2

$$\begin{aligned} & 6 \times 1024 && \text{(data directly indexed)} \\ & + 128 \times 1024 && \text{(data referenced by single indirect)} \\ & + 128^2 \times 1024 && \text{(data referenced by double indirect)} \\ & + 128^3 \times 1024 && \text{(data referenced by triple indirect)} \\ & = 2.02 \text{ GB} \end{aligned}$$

Inodes II

In a particular FS, an inode contains 6 direct pointers, 1 pointer to a (single) indirect block and 1 pointer to a doubly indirect block.

Each

byte

How

- 1 the 1020th data byte?
- 2 the 510100th data byte?

Inodes II

In a particular FS, an inode contains 6 direct pointers, 1 pointer to a (single) indirect block and 1 pointer to a doubly indirect block.

Each

byte

How

- 1 the 1020th data byte?
- 2 the 510100th data byte?

Our favourite numbers ☺

Answer: Inodes II

Assignment Project Exam Help

<https://eduassistpro.github.io>

Add WeChat edu_assist_pr

Answer: Inodes II

Assignment Project Exam Help

<https://eduassistpro.github.io>

Add WeChat edu_assist_pr

Answer: Inodes II

Assignment Project Exam Help

<https://eduassistpro.github.io>

Add WeChat edu_assist_pr

Answer: Inodes II

Assignment Project Exam Help

<https://eduassistpro.github.io>

Add WeChat edu_assist_pr

Answer: Inodes II

Assignment Project Exam Help

<https://eduassistpro.github.io>

Add WeChat edu_assist_pr

Assignment Project Exam Help

	Block	FAT	Inodes
Byt			ode only)
Byt		worst case	ode

Add WeChat edu_assist_pr

How do we manage a storage device's free space?

Need quick access to free blocks for allocation

Assignment Project Exam Help

Use `fr`

- <https://eduassistpro.github.io>
- Newly-freed blocks appended to end of list

Low overhead to perform free list maintenance

Add WeChat edu_assist_pr

Files likely to be allocated in noncontiguous blocks → increases file access time

Assignment Project Exam Help

<https://eduassistpro.github.io>

Add WeChat edu_assist_pr

Free List

Block size: 1 KB

Disk b

Num

requi

Hard drive size: 500 GB

Number of blocks: 488 million

Number of blocks required to store all addresses: 1.

$\frac{88}{255}$

Bitmap contains one bit (in memory) for each disk block

- Indicates whether block in use
- i^{th} bit corresponds to i^{th} block on disk

Adv

- <https://eduassistpro.github.io> in locations on secondary storage

Disadvantage

- May need to search entire bitmap to find free b in execution overhead

Assignment Project Exam Help

<https://eduassistpro.github.io>

Add WeChat edu_assist_pr

Assignment Project Exam Help

Bitmap

Block size: 1 KB

Hard

Num

Number of bits required: 488 million

Number of blocks required to store the bit map: $60,000$

Add WeChat edu_assist_pr

A possible file system layout

Assignment Project Exam Help

<https://eduassistpro.github.io>

Add WeChat edu_assist_pr

Fixed disk layout (with inodes)

- boot block
- superblock
- free inode bitmap

•

•

Super

- no of inodes
- no of data blocks
- start of inode & free space bitmap
- first data block
- block size
- maximum file size, ...

Directory

- Maps symbolic file names to logical disk locations (e.g. blah.txt → disk 0, block 2 (LBA))
- Helps with file organisation
- Ensures uniqueness of names

Assignment Project Exam Help

<https://eduassistpro.github.io>

Add WeChat edu_assist_pr

Assignment Project Exam Help

Single-level (or flat) file system

- Simplest file system organisation



<https://eduassistpro.github.io>

FS often performs linear search of directory to locate file

- Leads to poor performance

Add WeChat edu_assist_pr

Little flexibility in terms of file organisation

MultiLevel (Tree) Directory Structure

Hierarchical file system

- UNIX, Linux, Windows, Mac, ...
- **Root** indicates where on disk root directory begins
- **Root directory** points to various directories
 - Each of which contains entries for its files

Assignment Project Exam Help

<https://eduassistpro.github.io>

Add WeChat edu_assist_pr

Pathnames

- File names usually given as path from root directory to file

Absolute pathnames



<https://eduassistpro.github.io>

Relative

- Relative to working (or current) directory
- Can be changed using `cd` command
- Displayed with `pwd`
- Current directory: `.`
- Parent directory: `..`

Add WeChat [edu_assist_pro](#)

open/close	Open or close a directory
search	Find file in directory system using pattern matching on string wildcard characters
create/delete	Create or delete files/directories
link	
unlink	
chdir	
list	Lists or displays files in directory
read attributes	multiple read entry operation Read attributes of file
write attributes	Change attributes of file, e.g. permissions or name
mount	Creates link in directory to directory in different file system, e.g. on another disk or remote server

System Call	Description
<code>s = mkdir (path, mode)</code>	Create a new directory
<code>s = rmdir (path)</code>	Remove directory
<code>s = link (path, linkpath)</code>	Link path to linkpath
<code>s = unlink (path)</code>	Remove path
<code>s = chdir (path)</code>	Change current directory to path
<code>dir = opendir (path)</code>	Open directory path
<code>s = closedir (dir)</code>	Close directory dir
<code>dirent = readdir (dir)</code>	Read one entry from directory
<code>rewinddir (dir)</code>	Rewind directory to re-read

Assignment Project Exam Help

<https://eduassistpro.github.io>

```
struct dirent {
    long d_ino;           /* inode number */
    off_t d_off;          /* offset to next dirent */
    unsigned short d_reclen; /* length of this d_
    unsigned char d_type;   /* file type; not supported
                           /* by all file system types */
    char d_name [NAME_MAX+1]; /* file name (null-terminated) */
};
```

Steps in looking up `/usr/ast/mbox`

Assignment Project Exam Help

<https://eduassistpro.github.io>

Add WeChat edu_assist_pr

Link: Reference to directory/file in another part of FS

- Allows alternative names (and different locations in tree)

Assignment Project Exam Help

Hard link: Reference address of file

- Only supported for files in Unix

Sym <https://eduassistpro.github.io>

-

Problems

- File deletion: search for links and remove the
 - Leave links and cause exception when used (symbolic links)
 - Keep link count with file → delete file when count = 0 (hard links)
- Looping: directory traversal algorithms may loop

Assignment Project Exam Help

<https://eduassistpro.github.io>

Add WeChat edu_assist_pr

Mount operation

- Combines multiple FSs into one namespace
- Allows reference from single root directory
- Support for soft-links to files in mounted FSs but not

Mou

- Directory in native FS assigned to root of mou

FSs manage mounted directories with

- Information about location of mount poin
- When native FS encounters mount point, use mount table to determine device and type of mounted FS

Assignment Project Exam Help

<https://eduassistpro.github.io>

Add WeChat edu_assist_pr

Second extended file system (1993)

Goal: high-performance, robust FS with support for advanced features

Typical <https://eduassistpro.github.io>

Safety mechanism: 5% of blocks reserved for root

- Allow root processes to continue to run after user process consumes all FS disk space

Represents files and directories in ext2 FS

Stores information relevant to single file/directory → e.g. time stamps, permissions, owner, pointers to data blocks

ext2 inode pointers



<https://eduassistpro.github.io>



14th pointer is a doubly-indirect pointer



Locates block of indirect pointers



15th pointer is triply-indirect pointer



Locates block of doubly indirect pointers

Provides fast access to small files, while supporting very large files

Block groups

- Clusters of contiguous blocks
- FS attempts to store related data in same block group
- Reduces seek time for accessing groups of related data

Block

- <https://eduassistpro.github.io>
- FS was mounted, ...
- Redundant copies of superblock in some

Add WeChat edu_assist_pr

Inode table: Contains entry for each inode in block group

Inode allocation bitmap: Inodes used within block group

Block allocation bitmaps: Blocks used within group

Group descriptor: block numbers for location of:



<https://eduassistpro.github.io>

block allocation bitmap

accounting information

Add WeChat edu_assist_pro

Data blocks: Remaining blocks store file/dir



Directory information stored in directory entries

Each directory entry is composed of: inode number, directory entry length, file name length, file type, file name

Assignment Project Exam Help

<https://eduassistpro.github.io>

Add WeChat edu_assist_pr

Assignment Project Exam Help

Feature	ext2	ext3	ext4
Year	1993	2001	2008
Max file size	16 GB – 2TB	1	2.6.19
File system size	2 GB – 32 GB	2 GB	(te)

<https://eduassistpro.github.io>

Add WeChat [edu_assist_pro](https://eduassistpro.github.io)