

File Systems

All modern systems utilize an Operating System to facilitate the storage of data in units called “files”:

```
$ ls -la
```

drwxr-xr-x	5	waf	waf	4096	Oct 11 17:59	.
drwxr-xr-x	3	root	root	4096	Aug 31 15:15	..
-rw-----	1	waf	waf	9405	Oct 17 03:15	.bash_history
-rw-r--r--	1	waf	waf	3771	Aug 31 15:15	.bashrc
-rw-r--r--	1	waf	waf	65	Sep 9 16:01	.gitconfig
-rwxrwxrwx	1	waf	waf	5283	May 4 2020	may4.csv
drwxrwxrwx	1	waf	waf	4096	Apr 3 2021	mosaic
-rwxrwxrwx	1	waf	waf	6275	Mar 8 2021	mosaic.py

Permission Bits [1]	[3]	File Owner and Group [2]	File Size (bytes) [4] and Date Modified [5]	File Name [6]
---------------------	-----	--------------------------	---	---------------

[1]: Permission Bits:

d	r	w	x	r	w	x	r	w	x
Dir	User			Group			Other		

[2]: File Owner and File Group

- Specifies the user whom the user permission bits apply to. *(Each file can only have one user as the file owner.)*
- Specifies the group whom the user permission bits apply to. *(Each file can only have one group; each group may have many users, and users can belong to many groups.)*
- The owner of the file may give away the file ownership:
 - `chown <new owner> <file name>`
 - `chgrp <new group> <file name>`

[3]: File System Links (“Hard Links”)

[4]: File Size in Bytes

- The size of the contents of the file, in bytes. *(Note: even directories take up space!)*
- However, hard drive storage is **divided up into pages!** The actual storage capacity used may differ:

[5]: Last Modified Date:

- Almost all modern operating systems store three different date fields for every single file:
 - a.
 - b.
 - c.
- The date/time fields are always based on **your local computer clock** -- easily modified, easily faked.

[6]: File Name

- User-supplied name of the file.
- File Extensions:
- “dot” files and directories:

Q: Why does local file storage not work on a cloud-scale system?

Object Cloud Storage Systems

Instead of using local file storage, large data storage in the cloud-based systems are commonly stored as “objects”. These objects (files) are organized into _____:

Amazon AWS S3 CreateBucket REST API https://docs.aws.amazon.com/AmazonS3/latest/API/API_CreateBucket.html	
PUT / HTTP/1.1 Host: Bucket .s3.amazonaws.com x-amz-acl: ACL x-amz-grant-read: GrantRead : UserList x-amz-grant-write: GrantWrite : UserList x-amz-grant-full-control: GrantFullControl : UserList x-amz-grant-read-acp: GrantReadACP : UserList x-amz-grant-write-acp: GrantWriteACP : UserList [...]	
Bucket:	Name of the bucket. <i>[Required]</i>
ACL:	The canned Access Control to apply to the bucket. private public-read public-read-write authenticated-read
UserList:	You specify each grantee (user) as a type=value pair, where the type is one of the following: id – if the value specified is the canonical user ID of an AWS account uri – if you are granting permissions to a predefined group emailAddress – if the value specified is the email address of an AWS account Ex: x-amz-grant-read: id="11112222333", id="44445555666"
ACP:	x-amz-grant-read grants permission for the file itself; x-amz-grant-read-acp grants permissions for the access control policies.

Q: In what ways does this differ from file systems?

Any number of “objects” (files) may be **PUT** into a bucket:

Amazon AWS S3 PutObject REST API https://docs.aws.amazon.com/AmazonS3/latest/API/API_PutObject.html	
PUT /Key HTTP/1.1 Host: Bucket .s3.amazonaws.com x-amz-tagging: Tagging x-amz-acl: ACL x-amz-grant-full-control: GrantFullControl : UserList x-amz-grant-read: GrantRead : UserList x-amz-grant-read-acp: GrantReadACP : UserList x-amz-grant-write-acp: GrantWriteACP : UserList [...]	
Content-Length: ContentLength	
Body	
Key:	Object identifier (“file name”), must be unique per bucket. <i>[Required]</i>
Bucket:	Name of the bucket. <i>[Required]</i>
ACL+:	<i>(Same policies as in CreateBucket.)</i>
Tagging:	A key-value pair of tags associated with a specific object. Ex: tag1=value1&tag2=value2
Body:	The contents of the object is sent as the payload of the HTTP packet.

Q: Is there a directory structure similar to traditional file systems?

Q: In both traditional file systems and S3, names must be unique. However, tagging allows for multiple files to have the same tag. What design possibilities does this open up for us?

All Cloud Providers provide Object Storage:

[Amazon AWS](#)

[Microsoft Azure](#)

[Google GCP](#)