

HTML/CSS

- Starts with DOCTYPE
- Followed by html, head, ref sheets + title(name on the tab), body + headings, paragraphs...

HTML elements

- <head> to contain stuff like website data, usually not seen by user
 - <style> for internal styling
 - <link rel= href=> for external style sheet (e.g. CSS)
- name of link for hyperlink, name of link
- for image, used with below as
 - src for source image link
 - alt for alternate text, e.g. when image cannot be displayed, for reasons such as slow connection
- <body> for starting body of page
 - <h1> to <h6> for headings, biggest to smallest
 - <p> for paragraphs
- for ordered lists (with 1 2 3), for unordered lists
- for list items in ordered/unordered lists
- *
 for single line break within text, **does not have a closing tag**

Security

Confidentiality

- Info only accessible by authorised users
- Authenticate user before showing info
- Keep info flow between authorised
- Cryptography encryption
- Example: running programs to steal info from networks

Integrity

- Safeguarding the accuracy and completeness of information and processing methods, with cryptography possibly
- Not allow info to be modified by unauthorised
- Detection by audit trails or maths
- Example: modification of unauthorised information such as online ratings

Availability

- Ensure the authorised users can access the info whenever needed
- keep info flow between authorised
- Info only accessible by authorised users
- Example: DoS/DDoS attack

Types of security attacks in between info source and destination

- Interruption: stops info from getting to destination
 - Affects Availability
- Interception: takes a copy of the sent info in between the source and destination
 - Affects Confidentiality
- Modification: modifies the info from the source and returns it to the destination
 - Affects Integrity
- Fabrication: Creates fake info and sends to destination
 - Affects Authenticity

Security services

- Confidentiality, encryptions
- Authentication, tells who sent, created the info
- Integrity
- Non-repudiation, authenticate + ensure integrity, shows who has done what
- Access control
- Availability

Filesystem

Physical disk structure

- Platter
- Sector
- Track
- Cylinder

Chained allocation

- reads 1 block a time, so if a data is on the 7th block, program needs to read the same number of blocks to access the data
- Average number of blocks read to find a particular record = half of the number of blocks, i.e. 4.5 or 5 for 9 (as most OS can only read an entire block)

Contiguous allocation: as all space is allocated to a file consecutively on the disk, the program only has to read the file system 1 time to get any data

Big O Notations

$O(\log(n))$: Indexed

$O(n)$: Chained

$O(1)$: Contiguous

Example:

1024 block file and we want to read the 512th block:

Indexed = $O(\log(1024)) = 10$ blocks read

Chained = $O(1024) =$ up to 1024 blocks read (512)

Contiguous = $O(1) = 1$ block read

Most efficient Disk allocation algorithm: contiguous

Least efficient Disk allocation algorithm: chained

Unix

- Has a tree-like structure:
 - Root (/): highest level
- Configuration files usually stored in the directory “/etc”
 - For a particular user, usually in files and directories in the users ~home directory
- Command **grep** used for searching strings in files

Indexed file storage

- Stores files in blocks called inodes
- Each file/directory is reference by an inode
- Efficient use of space and fast to read blocks
 - All inodes are numbered
 - Special blocks/files on disk called a directory
 - Directories contain the names of files and the inode number
 - Tree structure

POSIX

- Portable Operating System Interface (for Unix)