Basic computing theory developed in the lecture and tutorial materials from week 1 to week 6.

WEEK 2: INTRODUCTION TO OPERATING SYSTEMS

What is an Operating System?

- An operating system is a piece of software that sits between all programs and the computer's hardware.
    - Manages your computer
    - Runs programs
    - Interface between user and hardware
    - Provides services to programs & users
    - Protects users and programs from each other

The Onion Model
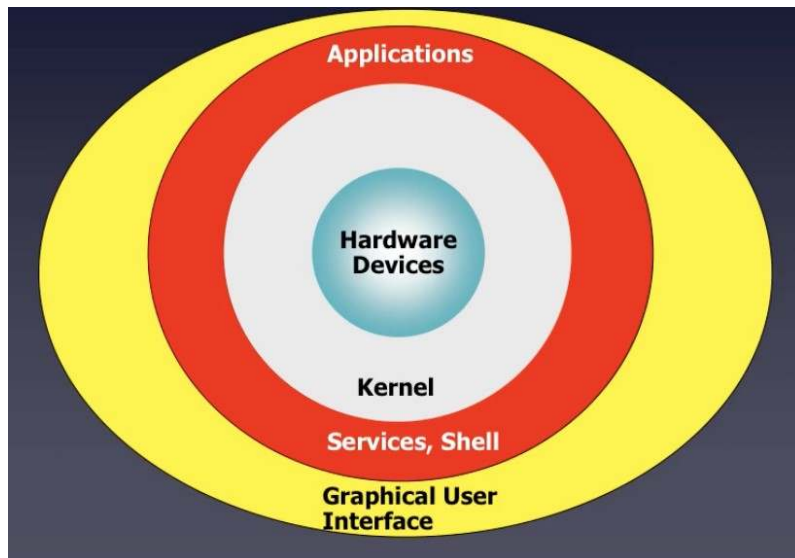
- Hardware: CPU, Memory, Input/output, storage.
- Kernel: controls the hardware directly, provides resources and services to application and manages access to privileged resources.
- Apps (userland): programs to do 'something' for the user and services are programs that run 'behind the scenes' (usually provides systems support), e.g. WIFI, security.
- Interface:
    - Shell (command line interface): a program that makes a set of command available to the user.
        - Interacts through the keyboard and a monitor which only prints text.
        - sh (1969), CPM (1973), cmd.exe
    - Graphical user interface: a user-friendly interface on top of the operating system, often runs the 'shell' commands transparently and sometimes a WEB interface.
        - *Interact via windows, icons, menu, pointer device (WIMP)*

Graphical user interface

Strengths	Weaknesses
- Little/no experience required - Good for graphics - User friendly, intuitive	- Can't do everything - Can crash the system - User is unsure of what the O/S is really

	doing
• Hides complexity from users	• Slows computer down • Needs better hardware • Hides complexity from users.

The Command Line Interface

Strengths	Weaknesses
• Greater flexibility • Fine tuning --> parameter • Essential for system administration • Faster, less overhead • Runs on simple hardware • Can run remotely • Robust- difficult to crash	• Hard to learn • Multiple options • Output often cryptic or non-existence • Inconsistent commands • No graphics • No safety net

Characteristics of a scripting language
- Variables are usually untyped = the same variable can be used as a number or a string
- Language syntax is often inconsistent
- Often designed and created by one person to get a particular job done
- Usually run through an interpreter, not a compiler

WEEK 3: OPERATING SYSTEMS II

UNIX history
- Many versions of UNIX but based of 2 original systems – System V and BSD.
- Used on most computers running the internet – web servers, domain names servers, emails servers, web hosting.

Standardising UNIX
- IEEE tried to standardise by defining commands, utilities, system interfaces, scripting language.
- POSIX has been largely ignored by vendors (too $$$ and complex).
- Resulting in inconsistency and difficulty in transferring code between systems.
- If version meets specs, it can be called UNIX, otherwise called 'UNIX-like'.

How has UNIX survived?
1. No one owns the ideas
2. UNIX is based on simple concepts
3. UNIX is portable
4. UNXI is mostly free
5. UNIX is efficient, stable and relatively secure
6. The UNIX as a set of tools approach.

File Systems
- A file system is a part of the operating system that manages data storage and access.
- Logical file system- user view of a file system
    - Files = Executable files (programs) and data files
    - Directories = store files and (usually) subdirectories, often hierarchical format
    - Partitions = Some directories may reside in different partitions from other directories, abstracts physical infrastructure from users.

- Physical file system- How these items are physically represented and stored.

Directories
- Create logical divisions in the file system
- For organisational purposes

Partitions
- Create physical divisions in the file system
- Can mount and unmount partitions
- Unmounting one partitions does not impact others
- Partitions are independent of each other.
    - Back up one without impacting others
    - Place dish quotas on one but not others

Theory of trees
- A tree is a collection of nodes along with a relation (parenthood)
- An edge is a 'branch' of the tree.
    - A = B means A is the parent of B
- Every node is a tree (expect the root) has exactly one parent. The root has no parent.
- A leaf is a node that has no children
- Siblings are node which have the same parent
- / = root of the file system
- /home = a Branch of the file system
- . = current directory (node)
- .. = parent of current directory

File systems and file manipulation
Disk physical structure. Organised into:
- Tracks: Concentric rings on a platter
- Heads: reads data from a platter.
- Cylinders: collection of all tracks on platters (which are horizontally in the same position).
- Sectors: part of a track for data.
- Platter

Disk Structure
- A dish is a stack of magnetic platters
    - This stack is divided into cylinders
    - Each cylinder contains circular tracks which are in turn divided into sectors
- Read/write operations are provided by the disk heads.
    - These move concurrently along the fixed disk arm
- The disk itself rotates with constant angular velocity to provide access to every sector.

Disk formatting
- Formatting is the operation which creates the physical disk structure
- Formatting is organizing and marking the surface of a disk into tracks, sectors and cylinders
- It is also sometimes a term to signify the action of writing a filesystem to a disk

Disk logical structures
- Partitions: disks can be subdivided into partitions - each is an independent storage device
- Blocks: the operating system views all the disk space as an array of fixed size logical blocks
    - o A logical block is the smallest unit of data to transfer.

File allocation
- Block: space is allocated to a file as one or more blocks
- Directory: is a table of information that the OS uses to locate blocks associated with flies on a disk.
- There are three common types of file allocation
    - <u>Contiguous</u> = is great for 'direct' storage & tine file systems O(1)
    - <u>Chained/Linked</u> = is good for archival O(n)
    - <u>Indexed</u> = is the only reasonable options for large systems O(log n)

<u>WEEK 4: THE WEB AND SECURITY</u>

Basic security principles

<u>Confidentially</u>
- Information is accessible only to authorised users:
    - o Can't be seen = encryption
    - o By whom? = Authentication
    - o When?
    - o Where? = Access Controls
    - o How? = Location, transmission path, protocol.

<u>Integrity</u>
- Safeguarding accuracy/ completeness of
    - o Information
    - o Processing methods
- Only entered/ altered by authorised users
- Cannot be altered without detection (in disk or over the web
- Detections
    - o Use audit trails
    - o Mathematical means: Hashes, Checksums, Message digests.

<u>Availability</u>
- Ensuring authorised users have access to information/processing when required
    - o Systems survive failures (have hot/cold standby mechanisms
    - o Systems resists attacks (Resistant to Denial of service (DoS) attacks)
- Users can access from authorised locations.

<u>Security service:</u> makes use of one or more security mechanism
- Confidentiality = privacy, encryption
    - o Encryption: converting plaintext into ciphertext to prevent non-intended recipients from reading.
    - o Most trivial crypto uses Symmetric key encryption.
    - o Problem is that the key needs to be secret and exchange between the parties involved in communication
- Authentication = who created or sent it
    - o Used for non-repudiation & access control
    - o Need to authenticate = people, organisation and application

- o E.g. something you know (password or pin), something you have (key, token, certificate or SMS), something you are (fingerprint, retinal scan, face recognition, voice recognition).
- o We can authenticate at the perimeter, at the operating systems and at the application.
- o Most sites use user id and password popup = basic = password is sent 'scrambled' but not encrypted
- Integrity = has not been altered
  - o Hashing is about putting a code on data to ensure it hasn't been modified by someone along the line or sent by someone else altogether.
  - o
    1. Do a checksum (modular sum of the characters in the file)
       cksum filename.txt
    2. Encrypt the checksum and sender's name with the sender's private key.
    3. Receiver uses the sender's public key to decrypt the checksum
    4. If error in the checksum, then the message has been modified along the way!
  - o Typically used in Secure email, Electronic documents and validate software.
- Non-repudiation = the order is final
  - o Can't deny that you signed for a transaction
- Access control = prevents misuse of resources
  - o Physical = tight control of physical access
  - o Logical = enforced by operating system, application, security devices
    - Needs configuration = management costs.
- Availability = permanence, non-erasure
  - o Ensuring authorised users have access to information/processing when required.
  - o System survive failures = have hot/cold standby mechanisms and backups
  - o Systems resist attacks - firewalls, CDN and Anti-malware

Security mechanism: designed to detect, prevent or recover from a security attack
Security Attack: an action that compromises the security of information

(a) Normal flow

(b) Interruption

(c) Interception

(d) Modification
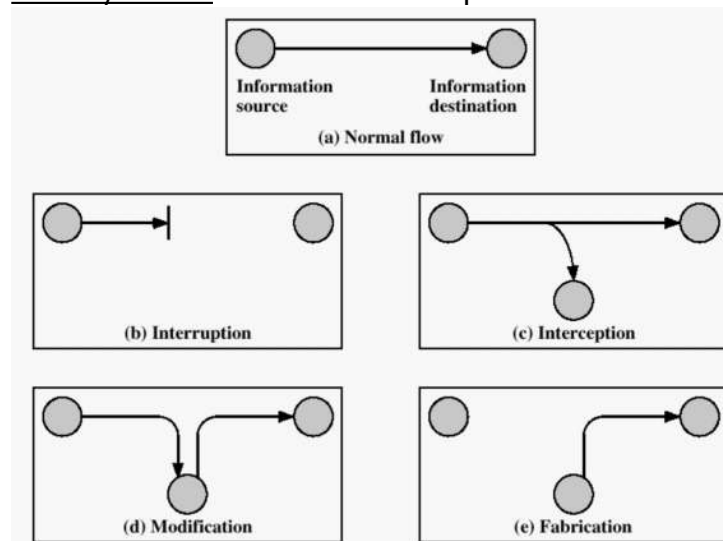
(e) Fabrication

- Normal flow: Information source --> Information destination

- Interruption: availability attack
- Interception: confidentiality attack
- Modification: integrity attack
- Fabrication: authenticity attack

Public Key Cryptography

- Each party has two keys (a private and public)
- Can encrypt with one and decrypt with the other
- Can be used for the four previously mentioned security capabilities.
  - Authentications - sender encrypts with their private key and receiver decrypt with sender's public key.
  - Privacy - sender encrypts with receivers public key.
  - Data integrity - it it's changed along the way, it can't be decrypted into anything meaningful
  - Non-repudiation - same reason as authentication.

Web Security: Encryption

- SSL (Secure sockets layer)
- Provides:
  - Confidentiality - stops interception
  - Integrity - stops modification
  - Authentication - Verifies owner of website and certificate based security
- Uses
  - Public key cryptography
  - Symmetric (shared secret) crypto

Unix security

- Typically by userid and password
- Trivial setups: saved in passwords file /etc/passwd
- Larger scale: Stored in central directory service

Access control

- 3 levels of security
  - User = owner of the file
  - Group = other users in the owner's group
  - Others = the public
- Each file and directory has 3 set of permissions
  - Read = can read the file
  - Write = can update the file
  - Execute = can execute (if a file) or traverse (if a directory)

Audit log

- Audit trails /logs are essential
- Needed to - Measure effectiveness, Do forensics, Create alters

Survive failures

- Good design and good operation
  - Standby mechanism
    - Hot standby = online system kept in sync
    - Cold standby = not-online but can be started quickly.
  - Redundancy = more than 1 server, data centre, network connection.
    - Protect against DDoS via network infrastructure.
  - Backups - on site and off site.

- o Firewalls - stop threats crashing your system = at the perimeter
- o Vaccinate - stop outages crashing your system
  - Install anti-malware
  - Monitor via security scanning systems
  - Training staff users
  - Protect against social-engineering

Security and risk assessments
- Security should match level of risk assessments
- Internal or external threat sources
- Vandalism
  - o Can be malicious
  - o Can be politically motivated
  - o Industrial espionage
  - o Theft

WEEK: THE WEB AND HCI

- HTML defines both structure and presentation of web pages
  - o E.g. structural and presentation tags.
- Structure: defines the components and areas on the page, e.g. top of page, middle section and footer.
- Presentation: Defines how the information is presented (style), e.g. colour, size, background, font.

CSS

- CSS: is a simple mechanism for adding style to web documents
  - o How we achieve the outcome of separating style from structure.
  - o CSS allows us to define the behaviours of each tag or container for out content.
  - o A set of rules defining how elements are to be presented. The Rules are set out as follows

Selector {
Property: value;
Property: value;
}

- 4 types of selectors available to us:
  - o Tag
    - This can be for any HTML tag
    - In stylesheet: p{color: yellow;}
    - In HTML: <p> this paragraph is yellow</p>
  - o #id
    - This can be for a particular element of our page.
    - Stats with a # symbol.
    - In stylesheet: #myYellow {color: yellow;}
    - In HTML: <p id="myYellow">this para is yellow</p>
  - o .class
    - Allows us to apply a style to a given group of elements
    - In stylesheet: .yellowPara {color: yellow;}
    - In HTML: <p class="yellowPara">this para is yellow</p>
  - o Inline

- Effectively hardcoding a style in an element without a stylesheet
- In HTML: <p style="color: yellow;">this para is yellow</p>
- Should avoid this technique where possible.
- To divide a page into sections with different styles:
  - <div>
    - Apply style to elements that need to be seperated from each other. Most browsers put a line break here.
  -
    - Apple a different style to part of your content inline without any line breaks.
    - E.g. this is inline styles.
- There are 4 fours ways in which CSS can be included in your page:
  - Imported- adding the contents of an external CSS file to another set of CSS rules.
    - Stylesheet: @import url(/css/mystyle.css)
    - HTML: only within <style> tags.
  - Linked- an external CSS file inclded in the head of the page (BEST WAY)
    - Stylesheet: this just contains the styles.
    - HTML: <link rel="stylesheet" href="/css/mystyle.css" type="text/css">
  - Embedded- including the CSS as a section in the head of the page.
    - HTML: <style> p ....
  - Inline- style included as an attribute of a tag
    - HTML: <p style="color:yellow;">
- CSS allows us to position HTML anywhere on a scren.
  - Can be relative to current position or absolutely places
  - Also specify margins, borders, spacing, padding
- Be awaee that styles cascade in the order of precedence
  - Browser default settings
  - User settings in brower
  - Linked External CSS
  - Imported CSS
  - Embedded CSS
  - Inline CSS
  - HTML Tag Attributes
- Nesting of tags
  - When tags are nested the styles for several tags may apply to a piece of content
  - Rule of thumb: inline before id before class & applied to innermost tag.

Benefits	Weaknesses
- A deinite seperation of style and structure - Simpler, cleaner code - Unifromity - The ability to define the look of several pages form one location - More power and flexibility than	- Not all browsers follow the standards properly - Quirks mode- where standards based browers try to emulate broken Internet Explorer

old methods • Better code resue	

-

1. Which of the following concept has simplified the conceptual design of UNIX?
Everything in UNIX are one of 'files', 'processes', 'permissions' and 'users'.

2. What programming language is the core of UNIX mostly written in?
C

3. Which of the following statements is true?
A 'cylinder' on a hard disk is a collection of tracks on all platters which are horizontally in the same position.

4. Which of the following 'Big O' notations has the LEAST cost?
O(1)

5. What is the main issue with Symmetric Key cryptography?
Secret key needs to be secret and exchanged between the 2 parties.

6. What sort of risk assessment would you consider if Mr Evil accesses the UTS 5-year strategic plans?
An industrial espionage external threat.

7. Which of the following best represents the colour RED in HTML?
#FF0000

8. What would the output of this HTML code look like?

```
<pre>To get a Distinction you need a
mark <HD and >C and & meet all the
requirements</pre>
```

To get a Distinction you need a mark C and & meet all the requirements.

9. Which CSS property will make all of the <p> content large size?
P {font-size: large;}

10. I have a file separate from my .html file that defines how the elements on HTML page should look like. What is this file?
An external CSS

1. I have a program which runs in the background and manages my network connection. In operating systems terminology, what typically is this program called?
Service

2. Which of the following is an advantage of using GUI?
They are more intuitive to use.

3. I am logged on as student and my current director is /home/student. What is the relative path to the Linuxgym-data director from this directory?
.../.../usr/local/linuxgym-data

4. A contiguous file systems has a file made up of 1000 blocks. How many times dfoes the fine systems have to be accessed to find the 27th block?
1

5. With regards to security, what does confidentiality mean?
Information is accessible only to the intended recipient.

6. Which type of security attack would it be if I were to fabricate your login to UTS online?
Authenticity Attack

7. What is the correct HTML code for making a hyperlink to a website?

website

8. which attribute in an <a> tag creates a ling to an anchor?

Href

9. Which CSS property is used to change the background color?

Background-color

10. If I am referring to a CSS selector which has properties and values then I am most likely talking about:

You cannot tell.