Information Technology Essentials — Lecture 06

Dr. Karim Lounis

Fall 2023

Software Systems

Software and Operating Systems

Since you did a Lab on Operating Systems, how do you define an OS?

Software and Operating Systems

You have certainly mentioned the concept of **Software**. So, what is a software?

Software

Definition

Software. It is a collection of programs, libraries, data, documentation as a single logical object to be run on a computer to perform specific tasks.

There are two different types of software

- **System Software.** Designed to provide the essential functions and services required for the operation of a computer.
- **Application Software.** Designed to perform specific tasks and functions for the user of a computer.

Note:

- An application is a software, but a software is not necessarily an application.

- Application software run **on top** of system software.

- A software that is designed to cause harm to computers is called a **malware**.

Software

Definition

Software. It i mentation as a single logica cific tasks.

There are two

- **System** unctions and services r
- **Applicati** ks and tions for

Note:

- An applicatio plication.
- Application so
- A software tha **malware**.

Program (A.k.a., Computer program). It is a finite sequence of instructions written in a programming language to solve a problem.

Ada Lovelace (1815-1852)

When a program is executing (it is running) on a computer, we call it a **process**.

A **process** is identified by a **PID** (Process ID).

Software

Definition

Softwar

a single

mentation as

cific tasks.

There ar

- **Sys**
  ser
- **Ap**
  tior

unctions and

ks and func-

Note:

- An app

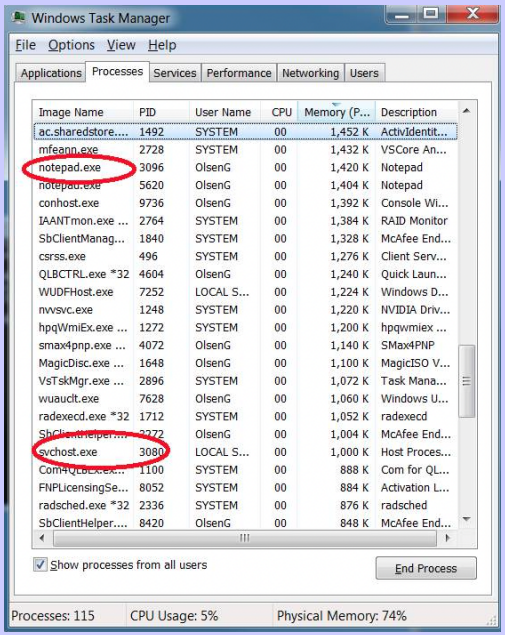
pplication.

- Applica

- A softw

malware.

On Windows PCs, you can use the command **taskmgt** to display the running processes (Windows task manager):



You can terminate a process by selecting it and hitting the "End Process" button (blocked processes, free space, ...).

Software

Definition

Softwar

a single

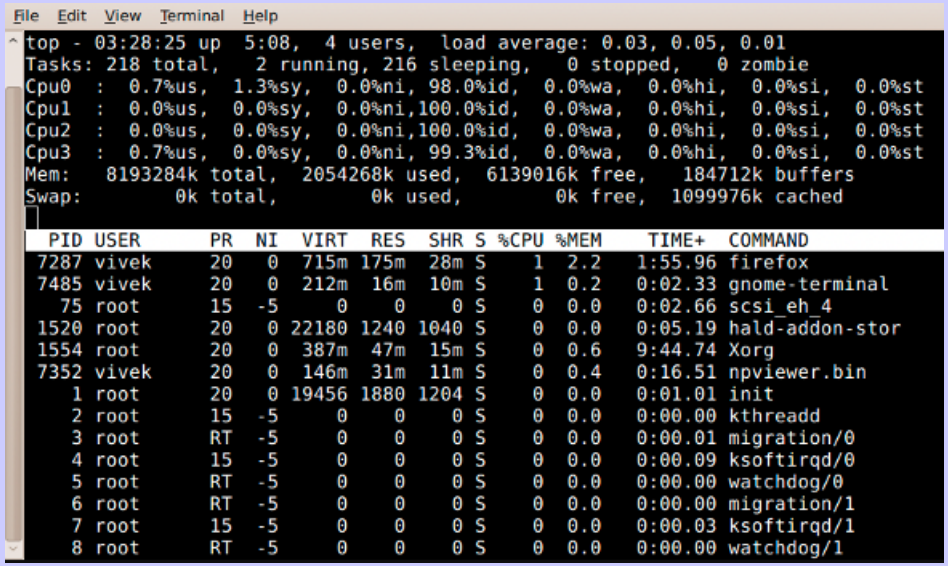
There ar

- **Sys**
  ser
- **Ap**
  tior

Note:

- An app
- Applica
- A softw

On GNU/Linux, you can use the command **ps** or **top** to display the running processes:



You can terminate a process by identifying their PID and running the command `kill -9 PID` on the terminal.

Software

Definition

Software. It i…n as a single logica…ks.

Program (A.k.a., Computer program). It is a finite sequence of instructions written in a programming language to solve a problem.

Ada Lovelace (1815-1852)

Some **processes** can be divided into multiple independent execution flows, called **threads** (lightweight process).

There are two …

- **System** …and services r…
- **Applicati**…unctions for t…

Note:

- An applicatio…
- Application sc…
- A software that is designed to cause harm to computers is called a **malware**.

Software

Definition

On GNU/Linux, you can type `ps -aux | grep "Google Chrome"` to display threads related to the Google Chrome browser:

```
1372 ??         0:02.39 /System/Library/Frameworks/ApplicationServices.framework/Frameworks/SpeechSynthesis.framework/Resources/com.apple.speech.speechsynthesisd
1390 ??         0:08.80 /System/Library/PrivateFrameworks/CoreLSKD.framework/Versions/A/lskdd
1467 ??         0:03.38 /System/Library/CoreServices/sharedfilelistd
1572 ??         0:24.34 /Applications/Google Chrome.app/Contents/Frameworks/Google Chrome Framework.framework/Versions/116.0.5845.187/Helpers/Google Chrome Helper (Renderer).app/Contents/MacOS/Google
1573 ??        29:37.86 /Applications/Google Chrome.app/Contents/Frameworks/Google Chrome Framework.framework/Versions/116.0.5845.187/Helpers/Google Chrome Helper (Renderer).app/Contents/MacOS/Google
1574 ??        53:21.01 /Applications/Google Chrome.app/Contents/Frameworks/Google Chrome Framework.framework/Versions/116.0.5845.187/Helpers/Google Chrome Helper (Renderer).app/Contents/MacOS/Google
1575 ??        15:30.98 /Applications/Google Chrome.app/Contents/Frameworks/Google Chrome Framework.framework/Versions/116.0.5845.187/Helpers/Google Chrome Helper (Renderer).app/Contents/MacOS/Google
1586 ??       128:41.49 /Applications/Google Chrome.app/Contents/Frameworks/Google Chrome Framework.framework/Versions/116.0.5845.187/Helpers/Google Chrome Helper (Renderer).app/Contents/MacOS/Google
1595 ??       156:14.74 /Applications/Google Chrome.app/Contents/Frameworks/Google Chrome Framework.framework/Versions/116.0.5845.187/Helpers/Google Chrome Helper (Renderer).app/Contents/MacOS/Google
1787 ??         9:35.72 /Applications/Google Chrome.app/Contents/Frameworks/Google Chrome Framework.framework/Versions/116.0.5845.187/Helpers/Google Chrome Helper (Renderer).app/Contents/MacOS/Google
1794 ??         0:37.19 /Applications/Google Chrome.app/Contents/Frameworks/Google Chrome Framework.framework/Versions/116.0.5845.187/Helpers/Google Chrome Helper (Renderer).app/Contents/MacOS/Google
1820 ??         0:06.54 /Applications/Google Chrome.app/Contents/Frameworks/Google Chrome Framework.framework/Versions/116.0.5845.187/Helpers/Google Chrome Helper (Renderer).app/Contents/MacOS/Google
2519 ??         0:02.55 /System/Library/PrivateFrameworks/CoreFP.framework/Versions/A/fairplayd
2581 ??         0:00.70 /System/Library/PrivateFrameworks/IMDPersistence.framework/IMAutomaticHistoryDeletionAgent.app/Contents/MacOS/IMAutomaticHistoryDeletionAgent
2969 ??         1:12.56 /System/Library/CoreServices/PowerChime.app/Contents/MacOS/PowerChime
2976 ??         3:36.54 /usr/sbin/distnoted agent
3058 ??         0:00.44 /System/Library/PrivateFrameworks/CommerceKit.framework/Versions/A/Resources/storedownloadd
3096 ??       512:57.69 /Applications/Adobe Acrobat Reader.app/Contents/MacOS/AdobeReader
3098 ??         1:39.33 /Applications/Adobe Acrobat Reader.app/Contents/Helpers/AcroCEF/RdrCEF.app/Contents/MacOS/RdrCEF "/Applications/Adobe Acrobat Reader.app/Contents/Helpers/AcroCEF/RdrCEF.app" --
3106 ??        58:24.75 /Applications/Adobe Acrobat Reader.app/Contents/Frameworks/RdrCEF Helper (GPU).app/Contents/MacOS/RdrCEF Helper (GPU) --type=gpu-process --log-severity=disable --user-agent-pro
3111 ??         8:18.74 /Applications/Adobe Acrobat Reader.app/Contents/Frameworks/AdobeCrashReporter.framework/Versions/A/Adobe Crash Handler.app/Contents/MacOS/Adobe Crash Handler 3096 AdobeReader 2
3113 ??         0:05.15 /Applications/Adobe Acrobat Reader.app/Contents/Frameworks/RdrCEF Helper.app/Contents/MacOS/RdrCEF Helper --type=utility --utility-sub-type=storage.mojom.StorageService --lang=
3116 ??         0:11.15 /Applications/Adobe Acrobat Reader.app/Contents/Frameworks/RdrCEF Helper.app/Contents/MacOS/RdrCEF Helper --type=utility --utility-sub-type=network.mojom.NetworkService --lang=
3118 ??         0:11.95 /Applications/Adobe Acrobat Reader.app/Contents/Frameworks/RdrCEF Helper (Renderer).app/Contents/MacOS/RdrCEF Helper (Renderer) --type=renderer --log-severity=disable --user-ag
3119 ??         0:50.04 /Applications/Adobe Acrobat Reader.app/Contents/Frameworks/RdrCEF Helper (Renderer).app/Contents/MacOS/RdrCEF Helper (Renderer) --type=renderer --log-severity=disable --user-ag
3130 ??         0:00.13 /System/Library/PrivateFrameworks/SystemStatusServer.framework/Support/systemstatusd
```

Threads from **Google Chrome** & from **Adobe Acrobat Reader** programs.

- A software that is designed to cause harm to computers is called a **malware**.

Software

Definition

Software. It i

a single logica

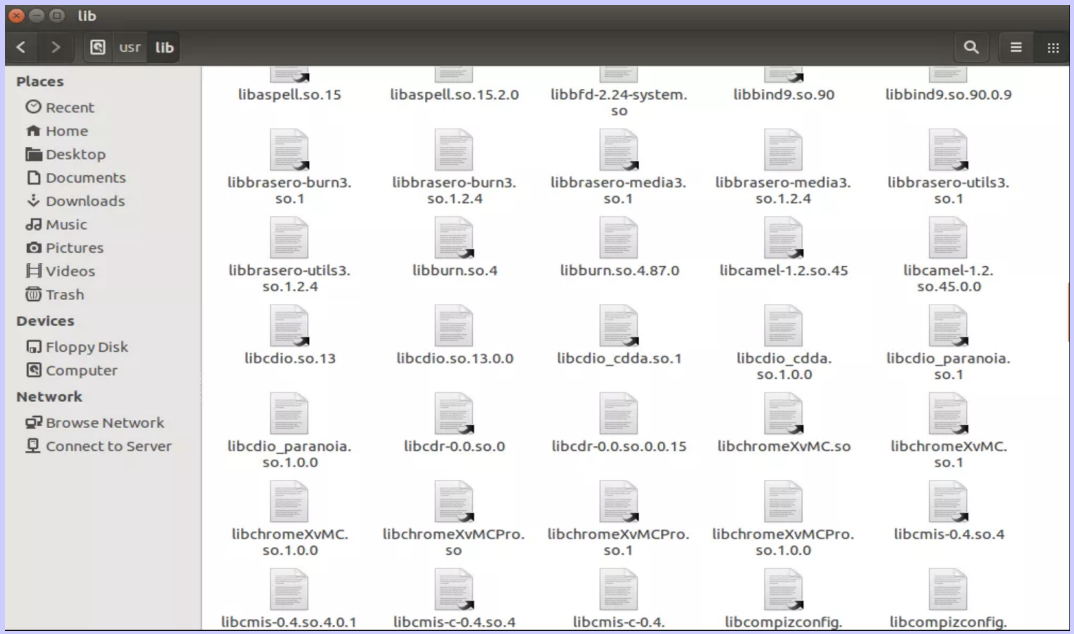
There are two

- **System**
  services r
- **Applicati**
  tions for t

Note:

- An applicatior
- Application sc
- A software tha

Library. Is a collection of reusable pre-written well-tested code (e.g., routines, functions, classes , modules, packages, etc), that a programmer can use to develop or run a software.



There are two types: **static libraries** and **dynamic libraries**.

Software

Definition

Softwar...ion as a single ...sks.

There a...

- **Sys**... ser... s and
- **Ap**... tior... func-

Note:

- An app...on.
- Applica...
- A softw...**re**.

There are two types: **static libraries** and **dynamic libraries**.

- **Static library.** A.k.a., archive libraries:
  - A file with extension *.a and *.lib.
  - Contains re-usable compiled object code.
  - The entire code is incorporated into the user's code.
  - Linking during compilation.
  - Increase program size but no compatibility issues.
- **Dynamic library.** A.k.a., shared libraries:
  - A file with extension *.so and *.dll.
  - Contains re-usable compiled object code.
  - The code is not incorporated into the user's code.
  - The library is loaded at runtime (on demand).
  - Linking during run-time (dynamic linking).

Software

Definition

Software. It i as a single logica

There are two

- **System** nd services r
- **Applicati** c- tions for

Note:

- An applicatio
- Application sc
- A software th

There is actually another type of libraries:

Standard libraries. These are collection of precompiled functions and procedures that are part of a programming language's standard ditribution.

They are provided as part of the compiler toolchain and are linked automatically during the compilation process. E.g.,:

- The `C` standard library (`libc`)
- The `C++` standard library (`libstdc++`).
- The `Java` standard library (includes lang, util, io, etc)
- The `Python` standard library (includes os, sys, etc)
- etc

Software

Definition

Software. It i tion as a single logica asks.

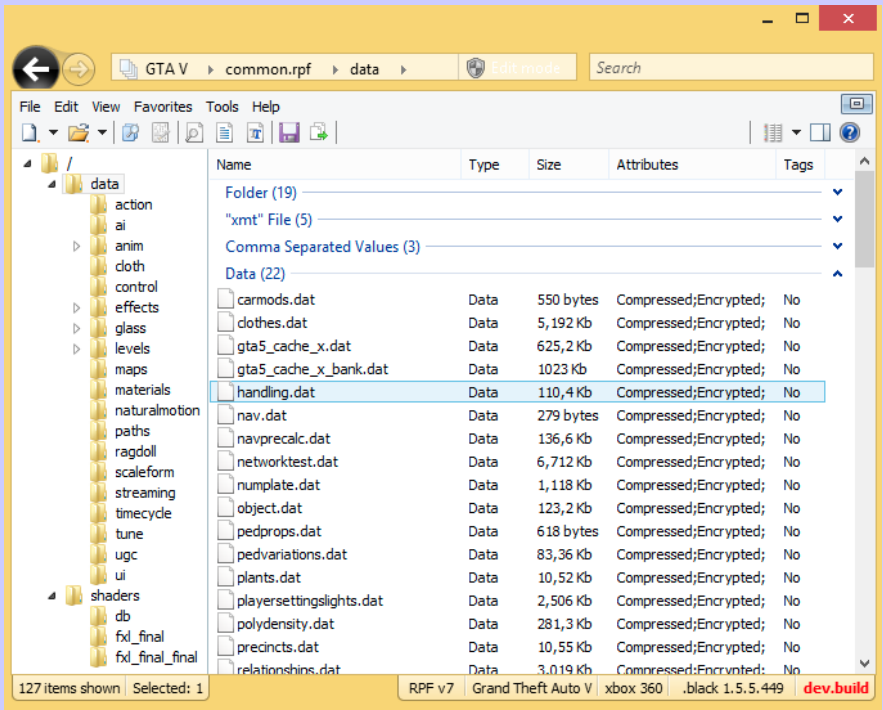
There are two

- **System** ns and services r
- **Applicati** func- tions for t

Note:

- An applicatio on.
- Application sc
- A software tha are.

Data. It refers to raw, unprocessed facts, figures, symbols, or values, e.g., numbers, text, images, audio, or video, generally pre-collected, recorded, or generated thru observations, measurements, or interactions.



Software

Definition

Software. It i tion as a single logica asks.

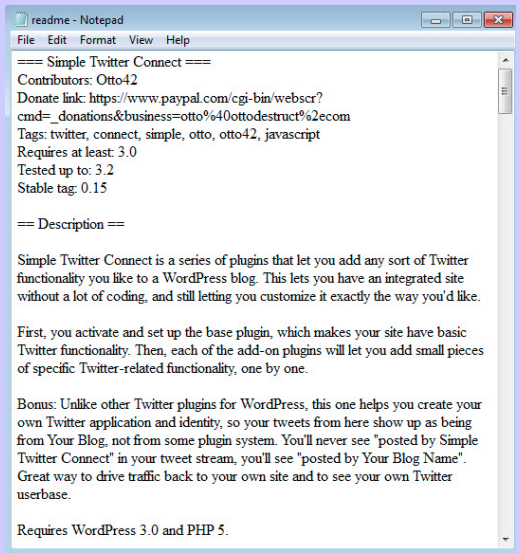
There are two

- **System** ns and services r
- **Applicati** func- tions for t

Note:

- An applicatior on.
- Application sc
- A software tha are.

Documentation It refers to a set of documents and files that aim to explain what the software is, how it can be installed, used, or should be used (terms and conditions).



Also, there are other files that may come with the software, such as configuration files.

Software

Definition

Software. It is a collection of programs, libraries, data, documentation as a single logical object to be run on a computer to perform specific tasks.

There are two different types of software

- **Application Software.** Designed to perform specific tasks and functions for the user of a computer.
- **System Software.** Designed to provide the essential functions and services required for the operation of a computer.

Note:

- An application is a software, but a software is not necessarily an application.

- Application software run **on top** of system software.

- A software that is designed to cause harm to computers is called a **malware**.

Software

- **Application Software.** Designed to perform specific tasks and functions for the user of a computer. Some examples (so there are more):
  - **Web browser.** E.g., Google Chrome, MS IE, Apple Safari, etc.
  - **Graphic processing tools.** E.g., Adobe Photoshop, paint, , etc.
  - **Word processing tools.** E.g., MS word, Libreoffice, Google Docs, etc.
  - **Spreadsheet tools.** E.g., MS Excel, Google Sheet, etc
  - **Media players.** E.g., VLC MP, WMP, iTunes, etc
  - **Email clients.** E.g., MS outlook, gmail, Thunderbird, etc.
  - **Gaming applications.** E.g., GTA, NFS, FIFA, etc.
- **System Software.** Designed to provide the essential functions and services required for the operation of a computer. Some examples:
  - **Operating systems.** E.g., MS Windows, GNU/Linux OS, MacOS, etc.
  - **Firmware.** E.g., OpenWRT, DD-WRT, TinyOS, etc.
  - **Device drivers.** E.g., Ralink 802.11n,
  - **Compilers.** E.g., gcc, g++, JRE, etc (sometimes application software)
  - **Utility Software.** E.g., disk cleaners, backup software, assembler, etc
  - **Middleware.** E.g., MySQL DBMS, web server (Apache), etc

Software Development LifeCycle

Software Development

What comes to your mind when it comes of software development?

Software Development

The development of software invovles:

1. Team work (cross-functional members).
2. Good planning and project management.
3. Adopt new programming languages, platforms, and frameworks.
4. Follow standard project management methodologies (e.g., scrum, xp, ...).
5. Decision making and customer satisfaction.
6. Commitement, accountability, and sustainabiity.
7. Ability of writing clean, well-documented, and efficient code.
8. Preserving users privacy, confidentiality, security, and safety.

Software Development Lifecycle — SDLC

The development of a software goes through various stages — these states are knowns **Software Development Lifecycle**.

These stages are as follows:

1. **Requirement Analysis.** Identify and understand the requirements and functionalities of the software.
2. **System Design.** Create a high-level system architecture that outlines the components of the software and their interaction. It includes designing the user interface (UI) and user experience (UX).
3. **Implementation.** A.a.k., coding. Here, the source code of the software is written using a selected programming language.
4. **Testing.** The developed software code is then tested (bugs hunting).
5. **Integration and Depeloyment.** A.k.a., operation. The subsystems are put together and the software is deployed for usage.
6. **Maintenance and Support.** Monitor the bahavior of the software, provide updates, fix flaws, and address user feedback.

Software Development Lifecycle — SDLC

The development of a software goes through various stages — these states are knowns **Software Development Lifecycle**.

These stages a

1. **Requiren** ts and functiona
2. **System I** utlines the comp es designing th
3. **Impleme** e software is w
4. **Testing.** The developed software code is then tested (bugs hunting).
5. **Integration and Depeloyment.** A.k.a., operation. The subsystems are put together and the software is deployed for usage.
6. **Maintenance and Support.** Monitor the bahavior of the software, provide updates, fix flaws, and address user feedback.

Step 1 — Requirement Analysis: In this step, the requirements (functional and non-functional) for the software are analysed. The purpose of requirement analysis is to ensure that the software works properly at the end and it is what the client asked for.

The approximate time to complete the software development is also estimated in this task. The output of this task is the software requirements specification.

Software Development Lifecycle — SDLC

The development of a software goes through various stages — these states are knowns **Software Development Lifecycle**.

These stages a

1. **Requiren** ts and functiona
2. **System I** utlines the comp es designing th
3. **Impleme** e software is w
4. **Testing.** The developed software code is then tested (bugs hunting).
5. **Integration and Depeloyment.** A.k.a., operation. The subsystems are put together and the software is deployed for usage.
6. **Maintenance and Support.** Monitor the bahavior of the software, provide updates, fix flaws, and address user feedback.

Step 2 — Design: Once the analysis for requirement is completed, the design process is started. This task builds the basic architecture of the project following which further development is made. It involves a software architect, software designer, UX and UI designer, database and security expert, etc.

One of the output of this phase, is the software design documents.

Software Development Lifecycle — SDLC

The development of a software goes through various stages — these states are knowns **Software Development Lifecycle**.

These stages a

1. **Requiren** ts and functiona
2. **System I** utlines the comp es designing the user interface (UI) and user experience (UX).
3. **Implementation.** A.a.k., coding. Here, the source code of the software is written using a selected programming language.
4. **Testing.** The developed software code is then tested (bugs hunting).
5. **Integration and Depeloyment.** A.k.a., operation. The subsystems are put together and the software is deployed for usage.
6. **Maintenance and Support.** Monitor the bahavior of the software, provide updates, fix flaws, and address user feedback.

Step 3 — Implementation: A.k.a., Development or programming phase. After the design process, the task of coding starts, which is also called as development process. The requirements and functionalities of the software are created in this stage.

Software Development Lifecycle — SDLC

The development of a software goes through various stages — these states are knowns **Software Development Lifecycle**.

These stages a

1. **Requiren** ts and functiona
2. **System I** utlines the comp es designing th
3. **Implementation.** A.a.k., coding. Here, the source code of the software is written using a selected programming language.
4. **Testing.** The developed software code is then tested (bugs hunting).
5. **Integration and Depeloyment.** A.k.a., operation. The subsystems are put together and the software is deployed for usage.
6. **Maintenance and Support.** Monitor the bahavior of the software, provide updates, fix flaws, and address user feedback.

Step 4 — Testing: Once the coding is done, testing is performed to ensure that the software is working correctly. This task assesses the software for errors and documents bugs if anything is present there in the coding. It is performed by a team of testers which uses known testing patterns.

Software Development Lifecycle — SDLC

The development of a software goes through various stages — these states are knowns **Software Development Lifecycle**.

These stages a

1. **Requiren** ts and functiona
2. **System I** utlines the comp es designing the user interface (UI) and user experience (UX).
3. **Implementation.** A.a.k., coding. Here, the source code of the software is written using a selected programming language.
4. **Testing.** The developed software code is then tested (bugs hunting).
5. **Integration and Depeloyment.** A.k.a., operation. The subsystems are put together and the software is deployed for usage.
6. **Maintenance and Support.** Monitor the bahavior of the software, provide updates, fix flaws, and address user feedback.

Step 5 — Deployment and Integration: A.k.a, operation step. It is the crucial task for running the developed software assessed by the stakeholders. This stage ensures that the developed software is running according to the customer's requirements.

Software Development Lifecycle — SDLC

The development of a software goes through various stages — these states are knowns **Software Development Lifecycle**.

These stages a

1. **Requiren** ts and functiona
2. **System I** utlines the comp es designing th
3. **Implementation.** A.a.k., coding. Here, the source code of the software is written using a selected programming language.
4. **Testing.** The developed software code is then tested (bugs hunting).
5. **Integration and Depeloyment.** A.k.a., operation. The subsystems are put together and the software is deployed for usage.
6. **Maintenance and Support.** Monitor the bahavior of the software, provide updates, fix flaws, and address user feedback.

Step 6 — Maintenance and Support: After the integration is done, the maintenance process occurs. It is a very necessary step for every software development. In this step, all the necessary upgrades and modifications are performed timely to add all the latest features in the software.

Software Development Lifecycle — SDLC

Developing software is not about writing, compiling, and executing code, then getting paid a big amount of money. It involves:

- Preserving names and reputation.
- Creating teams and cooperating with people you do not like, know, or even approach, or lazy + careless people (different level).
- Communicating and messaging over online platforms, e.g., Slack, MS Teams, Discord, etc (risk of misunderstanding & unprofessionalism).
- Managing projects using tools such as Asana, Basecamp, Jira, etc.
- Collaborative coding, using platforms such as github, bitbucker, etc.
- Developing soft-skills and respecting deadlines (commitments).
- Dealing with difficult situations and unexpected incidents.

Software (Exercise)

- How would you classify the virtual machine creator "Virtual Box"?
- How would you classify a linker and how would you classify a malware?
- Which among the following requires a lot of expertise and efforts to develop: hardware, software, firmware, middleware, or malware?
- How do we call the active version of a program (i.e., while the program is running on a computer)?
- When a program is executing, different sets of code can be run simultaniously. How we call those sets?
- What does it mean "`kill -9 4533`" in a GNU/Linux's terminal?
- Which part of the SDLC is considered crucial and may result in software development project failure?

Software, Free, Open-source, or blackbox

When companies or a group of people develop a software, the latter may be published as:

- **Proprietary Software.** This type of software is developed by a company and its source code is not disclosed (closed-source software). It is a blackbox — it provides the required services but does not show how. This type of software is usually not free. Maintenance is ensured.

  E.g., MS Windows, Apple's MacOS, Safari, MS Skype, GTA5, etc.

- **Open-source Software.** This type of software (OSS) released with a license that grants the public the right to view, use, modify, and distribute the source code of the software. It encourages collaboration and community involvement, allowing anyone to contribute to its development and improvement. They are free, but not always FREE.

  E.g., GNU/Linux OS, Mozilla firefox, libreOffice suite, GIMP, WordPress, VLC Media player, the Apache HTTP server, etc.

Software, Free, Open-source, or blackbox

When companies or a group of people develop a software, the latter may be published as:

- **Software Licenses:**
  It is a paper-based or digital proof that provides the user (or group of users) with rights to use the software because they purchased or got it for free from the author or software development company.

  E.g., MS Windows, Apple's MacOS, Safari, MS Skype, GTA5, etc.

- **Open-source Software.** This type of software (OSS) released with a license that grants the public the right to view, use, modify, and distribute the source code of the software. It encourages collaboration and community involvement, allowing anyone to contribute to its development and improvement. They are free, but not always FREE.

  E.g., GNU/Linux OS, Mozilla firefox, libreOffice suite, the Apache HTTP server, etc.

Software, Free, Open-source, or blackbox

When companies or a group of people develop a software, the latter may be published as:

- **Software Licenses:**
  Proprietary software are published under **restricted licenses** that outline terms and conditions for use. They often restrict how the software can be used (e.g., pay), modified, and distributed.

- Open-source software however, are generally published under **open-source licenses**, such as the GNU General Public License (GPL), Apache License, BSD License, Mozilla public license, Creative Commons license, and MIT License. They ensure that the software remains open and accessible to the community and that users have the freedom to collaborate, improve, and share the software.

E.g., GNU/Linux OS, Mozilla firefox, libreOffice suite, the Apache HTTP server, etc.

Software, Free, Open-source, or blackbox

When companies or a group of people develop a software, the latter may be published as:

- **Proprietary Software.** This type of software is developed by a company and its source code is not disclosed (closed-source software). It is a blackbox — it provides the required services but does not show how. This type of software is usually not free. Maintenance is ensured.

  E.g., MS Windows, Apple's MacOS, Safari, MS Skype, GTA5, etc.

- **Open-source Software.** This type of software (OSS) released with a license that grants the public the right to view, use, modify, and distribute the source code of the software. It encourages collaboration and community involvement, allowing anyone to contribute to its development and improvement. They are free, but not always FREE.

  E.g., GNU/Linux OS, Mozilla firefox, libreOffice suite, the Apache HTTP server, etc.

Ethics of Software Use

It is imporant to know the ethics related to using software:

- Respecting Software Licensing Agreements

  legal ramifications (lawsuits, fines, ...), reputation damage (destroy partnerships, future opportunities, loss of trust), financial consequences, limitation of software access (updates, patches, etc)

- Avoiding software piracy (cracked versions)

  negative impact on the software industry, risk of malware, legal penalties, ...

- Respecting Intellectual Property Rights

  Violating copyrights, patents, and tradmarks results in lawsuits

- Regular updates and patches installation

  issues related to security, bugs, performance, compatibility, etc

- Use software for good

  harm computers, people, company, community, country, etc

- End.