Computer Science Practice and Experience: Operating Systems 3SH3

Dr. Bojan Nokovic, P. Eng.

Based on: "Operating Systems Concepts", 10th Edition Silberschatz Et al.
"Intro Slides 3SH3 '12" - Sanzheng Qiao

Dec. 27, 2020

About 3SH3

- Course Outline
- `www.os-book.com`
- Virtual Box, GNU/Linux

Computer System Structure

A computer system can be divided roughly into four components:

1. Hardware
2. Operating system
3. Application programs
4. Users

What is an Operating System?

Computer System Structure

A computer system can be divided roughly into four components:

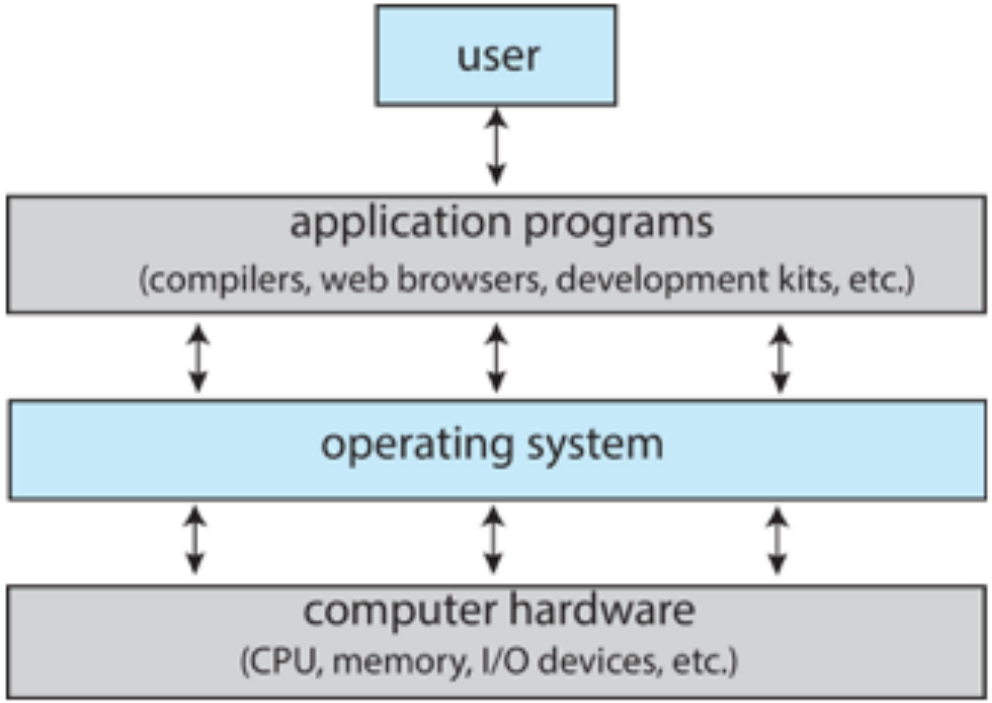
1. Hardware
2. Operating system
3. Application programs
4. Users

What is an Operating System?

OS controls and coordinates use of hardware among various applications and users.

Environment, government

Components of Computer



Why is Studying OS Interesting?

- Combining many things such as languages, hardware, data structures, algorithms.
- Creating illusions like a wizard making computer appear to be more than it really is.
  - Memory as large as disk
  - Single processor running multiple programs concurrently

OS Complexity

Microsoft's Windows OS, one of the most complex software tools ever built for a single computer is likely in the realm of 50 million lines.

There has never been a more interesting time to study operating systems!

Why is studying OS interesting?

- Managing large, complex systems (including users and programs), often in conflict needs (interactive vs. batch, friendly vs. secure, small jobs vs. big jobs).
- Protecting users from each other; security issues pose tough problems especially in networking environment.
- Presenting general techniques. So you can learn and apply them elsewhere.

Discipline and Craftsmanship

How to build large software systems in a reasonable amount of time, and make them work reliably?

Programming is a craft: something like a high-tech form of silversmithing.

To make big systems work, it takes a tremendous amount of discipline and structure. Unfortunately, many of us aren't too good at this stuff. Fortunately, if you are, you'll have a tremendous advantage.

Discipline and Craftsmanship

Rule 1: Simplicity. It's easy to make things complicated, harder to make them simple. Don't accept complexity.

Simplicity is the ultimate sophistication. - *Leonardo DaVinci*

Rule 2: Adopt a consistent style and use it everywhere. Decide on file organization, procedure structuring, naming conventions, location of curly braces, everything. By doing things consistently, they don't get omitted.

Success is neither magical nor mysterious. Success is the natural consequence of consistently applying basic fundamentals. - *E. James Rohn*

Discipline and Craftsmanship

Rule 3: Don't litter. Temptation is to make fast fixes that dirty things up. It's crucial to take the time to fix things right at the beginning. You'll never have time to come back to it later!

Discipline and Craftsmanship

Rule 4: Document carefully as you go. Don't put this off! Most important things are interfaces: procedure headers and data structures and other things that tie together the parts of the system.

Rule 5: Documentation should describe things at a higher level than code.

Rule 6: Put documentation near the code: otherwise you forget to change the documentation when you change the code.

Rule 7: Quality, not quantity!

IEEE Code of Ethics

Rule 1: To hold paramount the safety, health, and welfare of the public, to strive to comply with ethical design and sustainable development practices, and to disclose promptly factors that might endanger the public or the environment;

Rule 2: To avoid real or perceived conflicts of interest whenever possible, and to disclose them to affected parties when they do exist;

Rule 3: To be honest and realistic in stating claims or estimates based on available data;

Rule 4: To reject bribery in all its forms;

IEEE Code of Ethics

Rule 5: To improve the understanding by individuals and society of the capabilities and societal implications of conventional and emerging technologies, including intelligent systems;

Rule 6: To maintain and improve our technical competence and to undertake technological tasks for others only if qualified by training or experience, or after full disclosure of pertinent limitations;

Rule 7: To seek, accept, and offer honest criticism of technical work, to acknowledge and correct errors, and to credit properly the contributions of others;

IEEE Code of Ethics

Rule 8: To treat fairly all persons and to not engage in acts of discrimination based on race, religion, gender, disability, age, national origin, sexual orientation, gender identity, or gender expression;

Rule 9: To avoid injuring others, their property, reputation, or employment by false or malicious action;

Rule 10: To assist colleagues and co-workers in their professional development and to support them in following this code of ethics.

Example Ethics

Consider the following situation: Suppose that you are a development engineer with responsibility for a software system employed in one of your company's major products. You seek to improve the efficiency of your software system and, following some research, you discover an algorithm posted on the Web that would provide a vast improvement for your system. The algorithm is written in the same language as that used by your system.

Q: Would it ever be ethical to copy the code that implements the algorithm and incorporate it in your system ?

Example Ethics

Consider the following situation: Suppose that you are a development engineer with responsibility for a software system employed in one of your company's major products. You seek to improve the efficiency of your software system and, following some research, you discover an algorithm posted on the Web that would provide a vast improvement for your system. The algorithm is written in the same language as that used by your system.

Q: Would it ever be ethical to copy the code that implements the algorithm and incorporate it in your system ?
A: Yes, provided you have obtained permission from the author and that you include an appropriate attribution to the author.

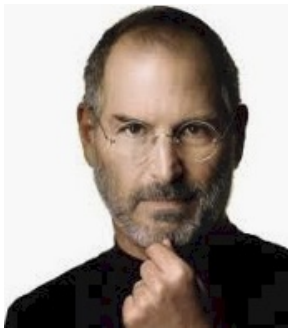
Popular Operating Systems

GNU/Linux - some distributions free, others open-source only.
`http://www.gnu.org/distros/`

Microsoft Windows - closed-source, proprietary software.

macOS - hybrid approach, based on open-source kernel named Darwin, but includes proprietary, closed-source components as well.

Bill Gates
Windows



Steve Jobs
macOS

Richard Stallman
GNU

Linus Torvalds
Linux

Open Source and Free Software

What is the difference between free OS and open-source OS?

Free Software

Source code available, licensed to allow no-cost use, redistribution, and modification.

Open Source

Does not necessarily offer licensing.

GNU/Linux History

Launched in 1983, by 1991, the GNU project had developed compilers, editors, utilities, libraries, and games

In 1991, a student in Finland, Linus Torvalds, released UNIX-like kernel using the GNU compilers and tools and invited contributions worldwide.

GNU/Linux operating system

Major distributions include Red Hat, SUSE, Fedora, Debian, Slackware, and Ubuntu.

Berkeley Software Distribution (BSD) UNIX

UNIX 1970, Bell Labs research center.

BSD UNIX started in 1978 as a derivative of AT&T's UNIX.

Open-source version, 4.4 BSD-lite, was released in 1994.

BSD UNIX distributions FreeBSD, NetBSD, OpenBSD, and DragonflyBSD.

Suggestions

Attend lectures and tutorials (skipping lectures can be costly, instead of saving you time)

Review lectures regularly, at least once a week (easier when it is still fresh, for the same reason, deferred exam is harder for you) - Know what you do not know !

Work independently (helps you understand the details)

Get help whenever it is needed.

The End of the Beginning

Let's work together.

Good Luck!