

Operating Systems: Introduction

Gabriel Laskar <gabriel@lse.epita.fr>

Outline

- What is an OS
- History
- Why do we need operating systems
- OS taxonomy (types and families)
- Kernel/Services/Applications

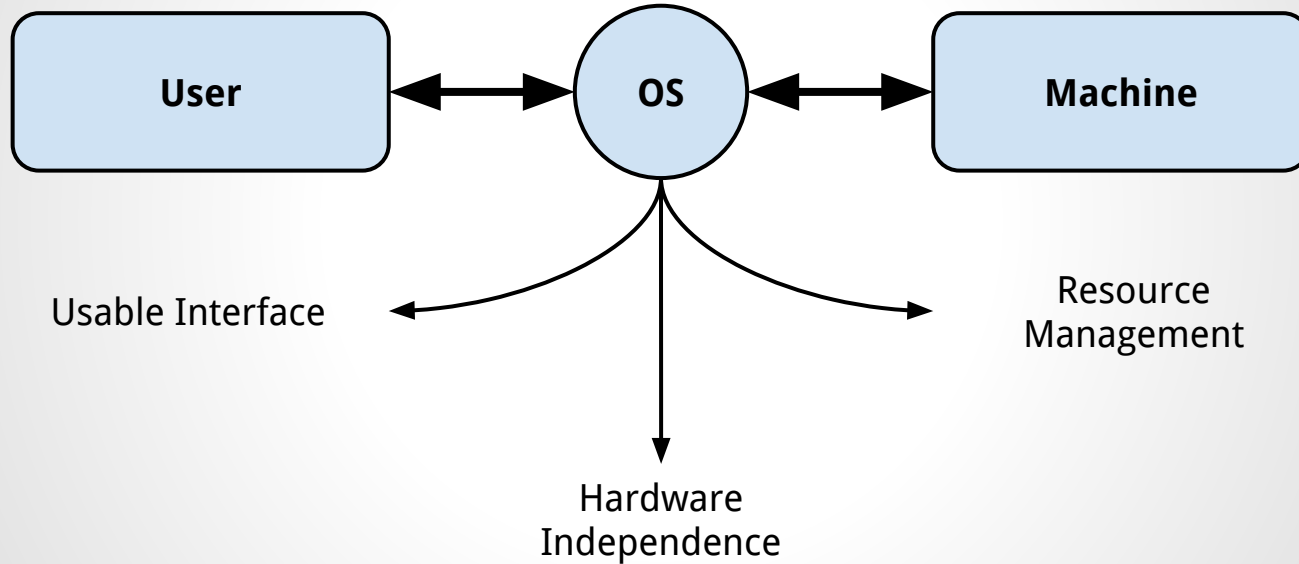
What is an Operating System?

Definition

An operating system (OS) is system software that manages computer hardware and software resources and provides common services for computer programs. The operating system is a component of the system software in a computer system. Application programs usually require an operating system to function.

-- Wikipedia

Another definition



Most used definition:

- Resource allocation
- Resource management

Other:

- first software layer on bare hardware
- only permanent software on a computer
- software that run in kernel mode

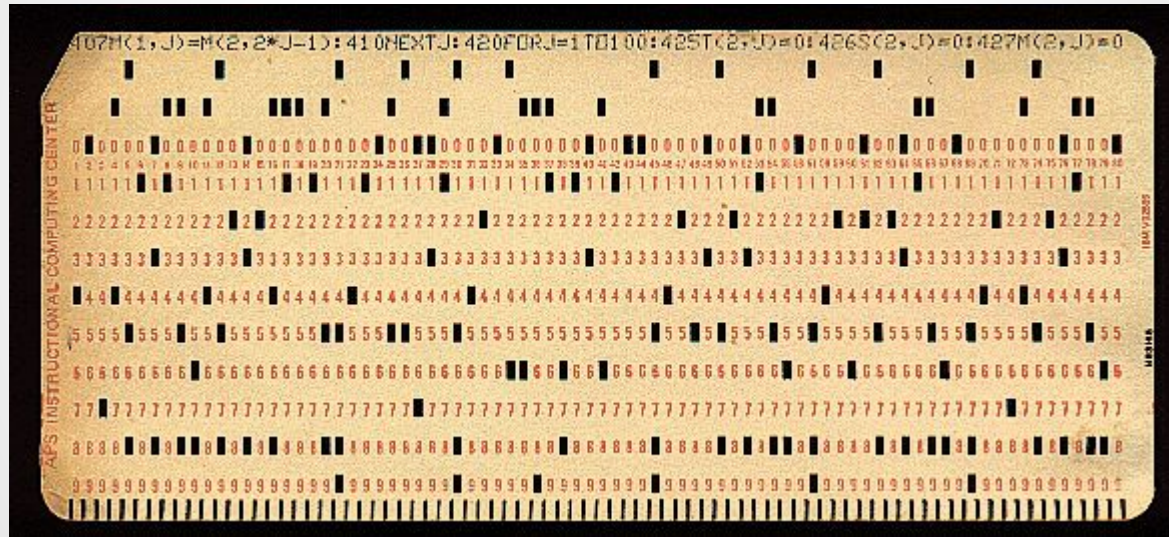
Where do we need an Operating System?

- computer
- mobile phone
- voip phone
- sim cards
- printer
- car
- ...

History

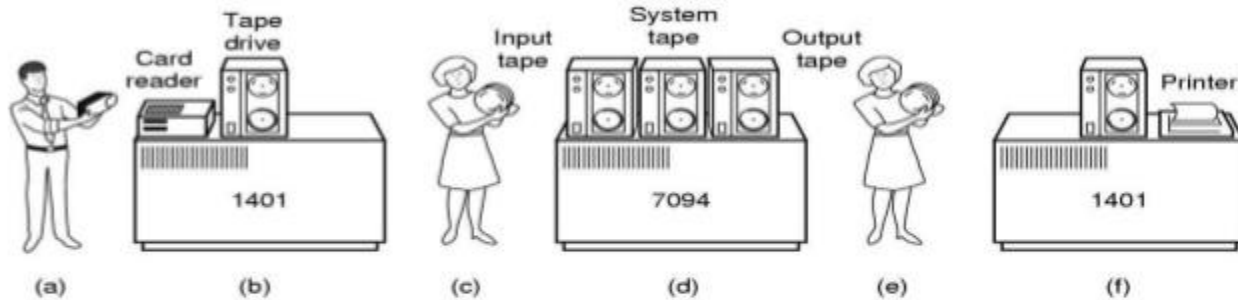
- generations
- unix
- windows
- personal computers

First Generation (1945-55)



Second Generation (1955-65)

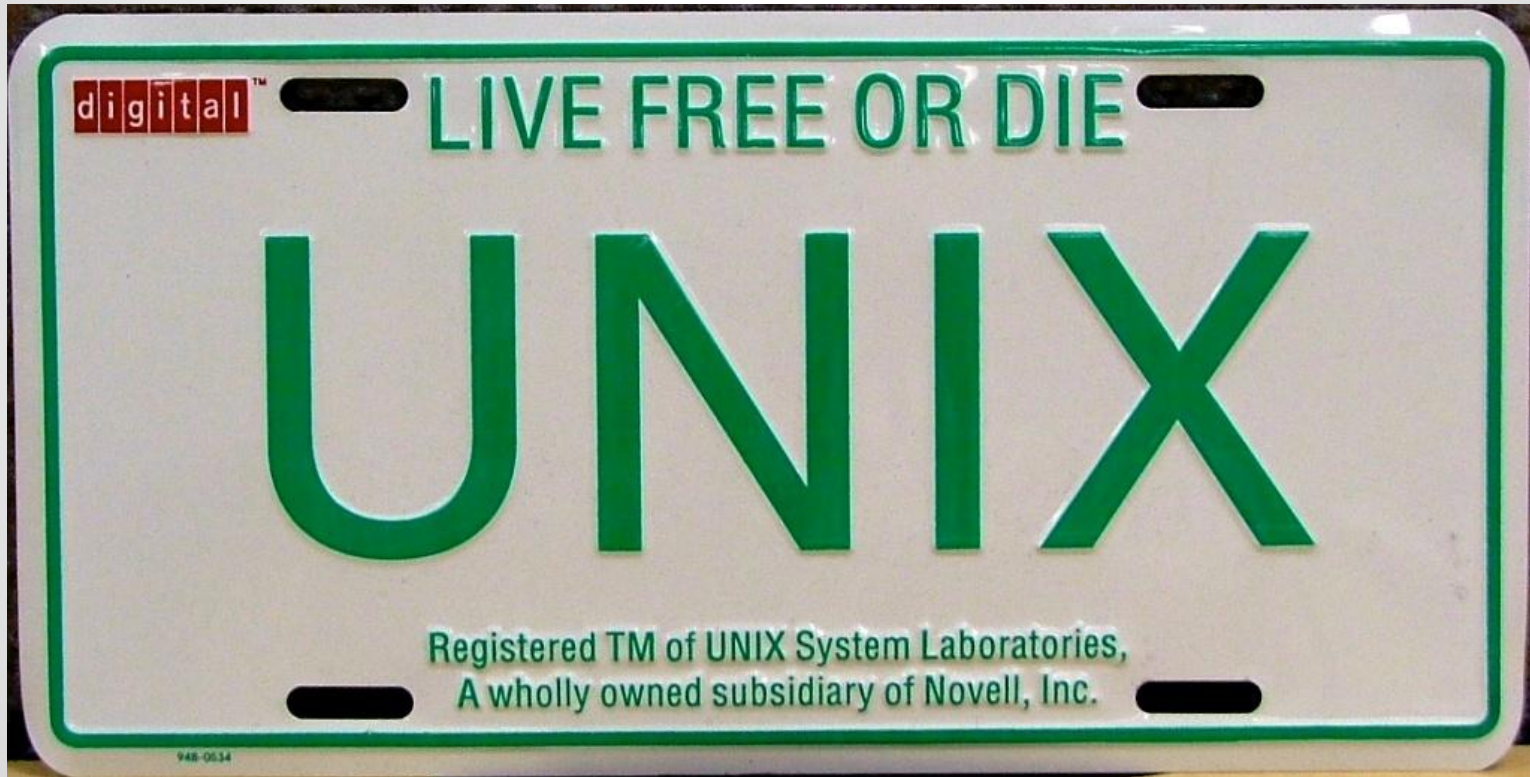
Evolution of Operating Systems (1)



Early batch system

- bring cards to 1401
- read cards to tape
- put tape on 7094 which does computing
- put tape on 1401 which prints output

Third Generation (1965-1980)



948-0534

Fourth Generation



OS taxonomy

It is difficult to split an operating system into a single category. Historically, each specific task has a specific operating system tailored for it. Now the barriers between the different workloads are blurred.

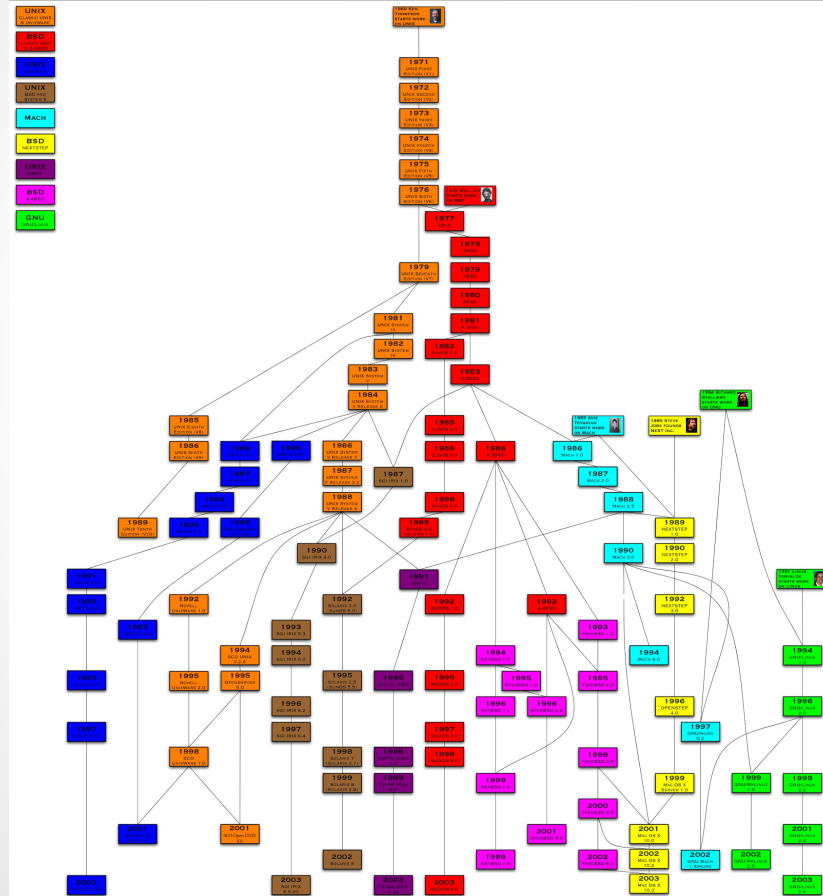
History: OS for a specific workload

- Mainframe
- Server
- Multiprocessor
- Personal Computer
- Embedded (ie Mobile)
- Real-Time (Soft or Hard)
- Smart Card

Mainframe Operating System



Server Operating System



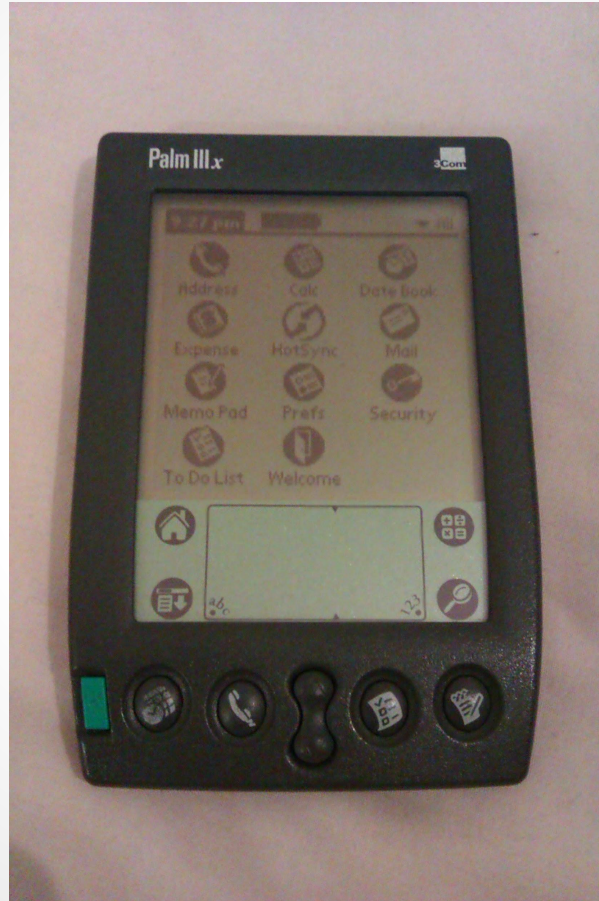
Unixes Jungle

- 1969 - 1979: Unix 1-6
- 1978 : BSD
- 1992 : FreeBSD, 4.4 BSD, NetBSD
- 1994 : OpenBSD
- 1987 : NextStep
- 2001 : Mac OS X
- 1987 : Minix
- 1991 : Linux

Personal Computer

- application driven
- multimedia
 - screen
 - graphic acceleration
 - input devices (keyboard, mouses, ...)
 - multiple kind of devices hooked on it

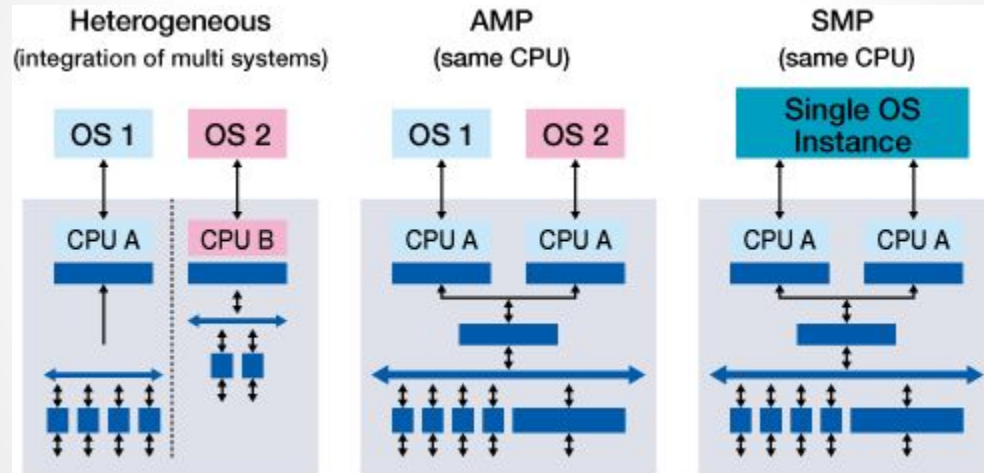
Embedded Systems



History: Why that much differences?

- computing power was limited
- for each workload a different way to handles tasks
 - Single-Task, Single-User
 - Multi-Task, Single-User
 - Multi-Task, Multi-User
- more computing power, more complexity forces OS designers and vendor to simplify and reuse more components

Multiprocessor Operating System



History: What is the actual status

- Real Time Operating Systems
 - Hard Real-Time
 - Soft Real-Time
- General purpose Operating Systems:
 - desktop (MS Windows, Apple OS X, Linux, Chrome OS)
 - server (Linux, MS Windows, Apple OS X)
 - mobile (Android, Apple iOS, Windows Phone)
 - embedded (Linux on multiple kind of devices, cars, cameras, printers...)
 - multiprocessor (most of the general cpus we find are now multi-core at least)
 - mainframe even (think s/390 with Linux/KVM)

What is a kernel?

- software that run in privileged mode
- “heart” of the system
- critical part (no error allowed inside)
- deliver the first abstractions
 - hardware independence
 - basic resource management
- This is a jungle too

Kernel design: multiple models

- monolithic kernel
- microkernel
- hybrid/modular kernel

- nanokernel
- exokernel
- unikernel

What is inside an OS?

Inside the OS

- Kernel: basic functionalities, syscalls
- Services: Accounts (login, nsswitch), Display Service
- Libraries: support functions, APIs (libc, libm, OpenGL)
- Applications: Web browser, Text Editor
- Support Applications: Terminal Emulator