Operating Systems
COMS(3010A)
Introduction

Branden Ingram

branden.ingram@wits.ac.za

CONTENT

- Introduction
- Abstraction/Virtualisation
- Concurrency
- Scheduling
- Persistence

Why is COMS3010 important

- Some of you will actually design and build operating systems or components of them.
    - Perhaps more now than ever

Why is COMS3010 important

- Some of you will actually design and build operating systems or components of them.
  - Perhaps more now than ever

- **Many of you will create systems that utilize the core concepts in operating systems.**
  - **Whether you build software or hardware**
  - **The concepts and design patterns appear at many levels**

Why is COMS3010 important

- Some of you will actually design and build operating systems or components of them.
  - Perhaps more now than ever

- Many of you will create systems that utilize the core concepts in operating systems.
  - Whether you build software or hardware
  - The concepts and design patterns appear at many levels

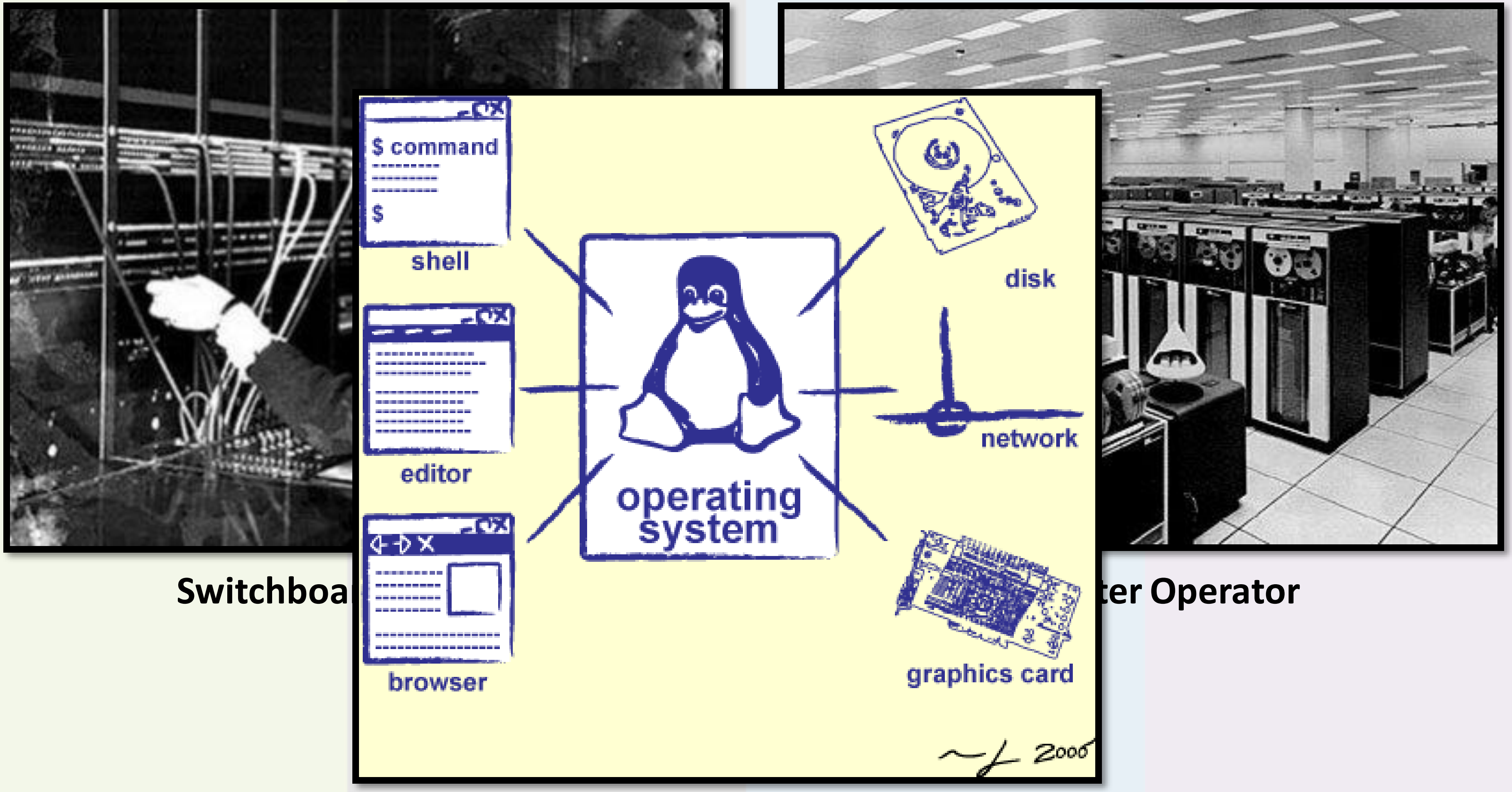
- **All of you will build applications, etc. that utilize operating systems**
  - **The better you understand their design and implementation, the better use you'll make of them.**

OS ? What is it ?

Switchboard Operator

Computer Operator

OS ? What is it ?



Switchboa

ter Operator

Software Operator

OS ? What is it ?

- OK, well what is a system then

OS ? What is it ?

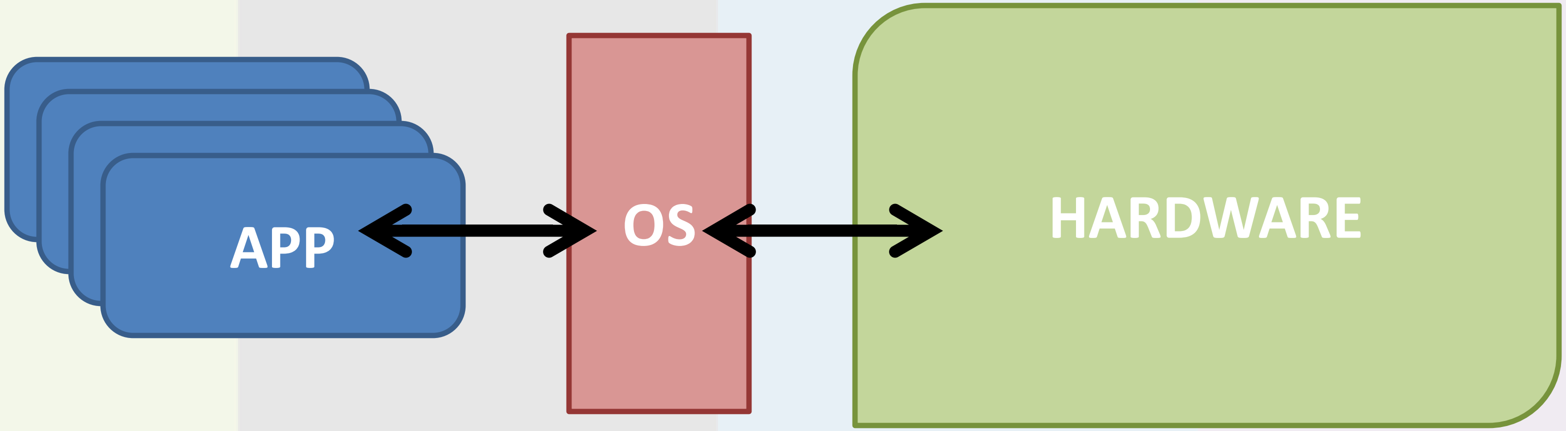
- OK, well what is a system then?

- **A set of things working together as parts of a mechanism or a interconnecting network**

OS ? What is it ?

- **Special layer of software that provides application software access to hardware resources**

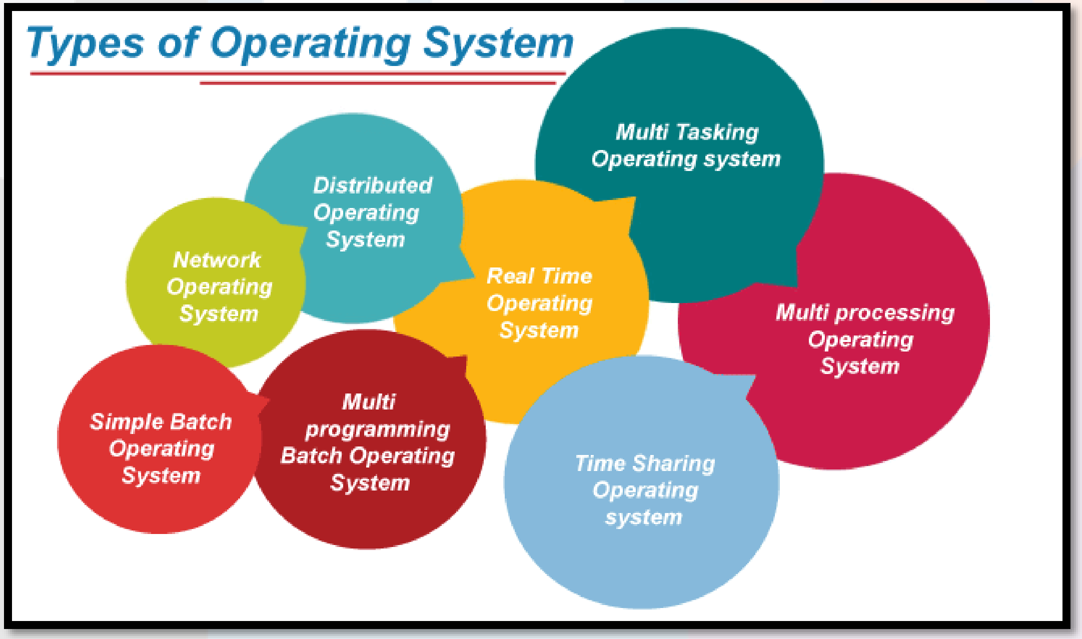
  - Convenient abstraction of complex hardware devices
  - Protected access to shared resources
  - Security and authentication
  - Communication amongst logical entities



Types of OS

- **An operating system is a well-organized collection of programs that manages the computer hardware.**

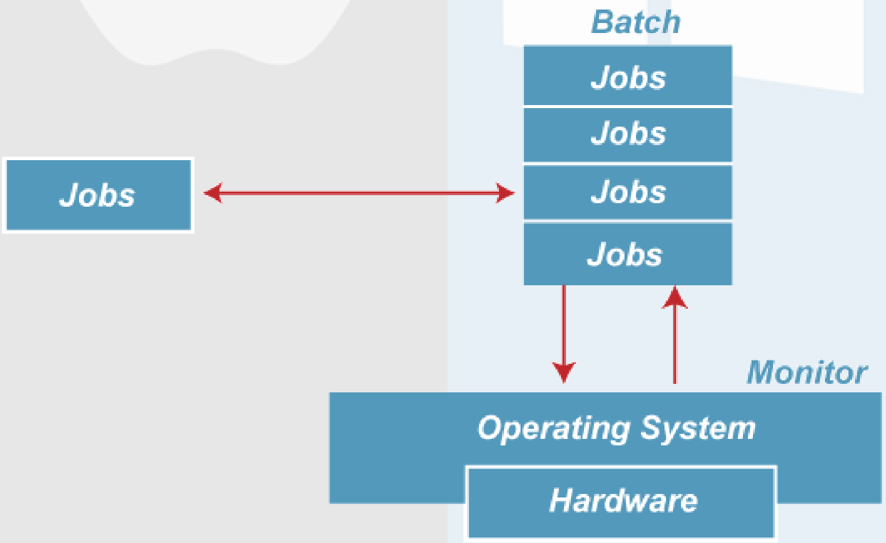
- Batch
- Embedded
- Multi-programming
- Multi-processing
- Distributed


Types of Operating System
Multi Tasking Operating system
Distributed Operating System
Network Operating System
Real Time Operating System
Multi processing Operating System
Simple Batch Operating System
Multi programming Batch Operating System
Time Sharing Operating system

Types of OS

- **Batch Operating System**

- Multiple users submit jobs which are queued and processed sequentially.
- These jobs then return output back to their respective user.



Types of OS

Advantages of Batch OS

1. The use of a resident monitor improves computer efficiency as it eliminates CPU time between two jobs.

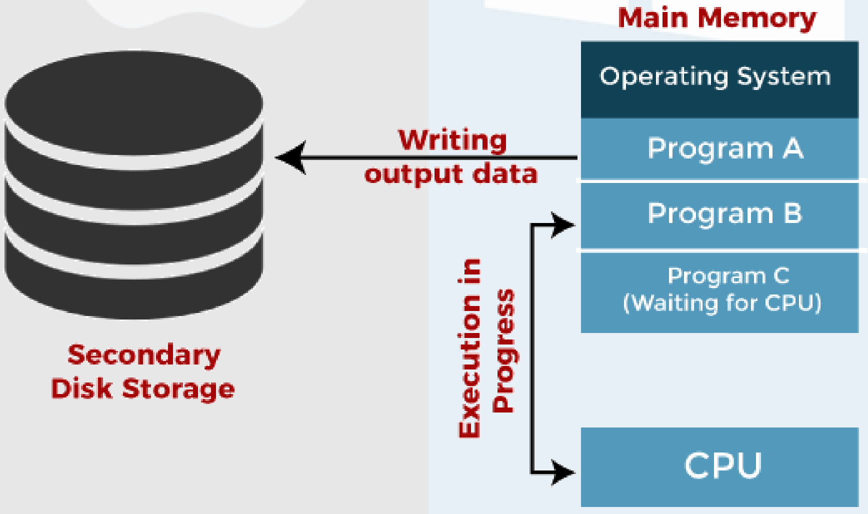
Disadvantages of Batch OS

1. Starvation
2. Not Interactive

Types of OS

- **Multiprogramming Operating System**

- Each process needs two types of system time: CPU time and I/O time.
- In a multiprogramming environment, when a process does its I/O, The CPU can start the execution of other processes.



Jobs in multiprogramming system

Types of OS

Advantages of Multiprogramming OS

1. Throughput is increased as the CPU always had one program to execute.
2. Response time can also be reduced.

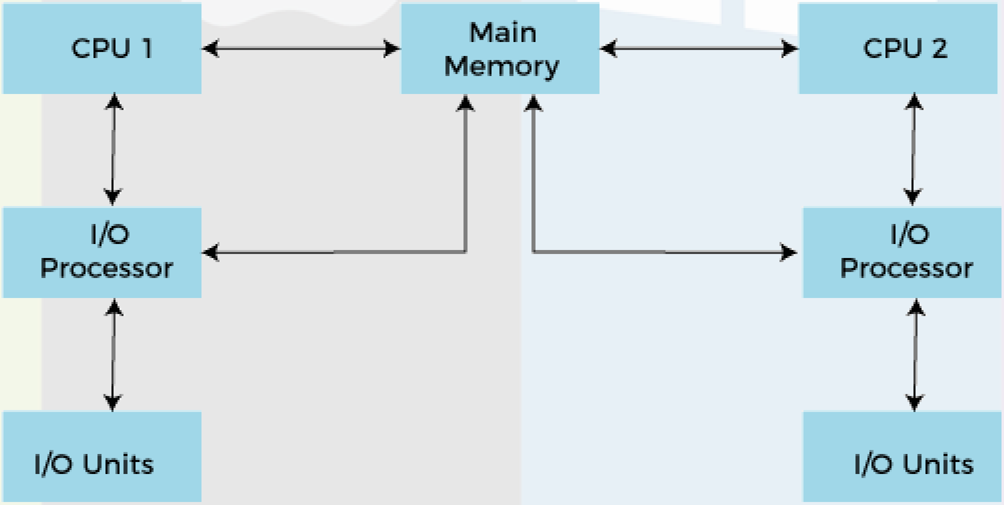
Disadvantages of Multiprogramming OS

1. Multiprogramming systems provide an environment in which various systems resources are used efficiently, but they do not provide any user interaction with the computer system.

Types of OS

- **Multiprocessing Operating System**

- Parallel computing is achieved. There are more than one processors present in the system which can execute more than one process at the same time.



Working of Multiprocessor System

Types of OS

Advantages of Multiprocessing operating system:

1. Increased reliability: Due to the multiprocessing system, processing tasks can be distributed among several processors. This increases reliability as if one processor fails, the task can be given to another processor for completion.
2. Increased throughput: As several processors increase, more work can be done in less.

Disadvantages of Multiprocessing operating System

1. Multiprocessing operating system is more complex and sophisticated as it takes care of multiple CPUs simultaneously.

Types of OS

- **Embedded Operating System**

- It is the specific purpose operating system used in the computer system's embedded hardware configuration.

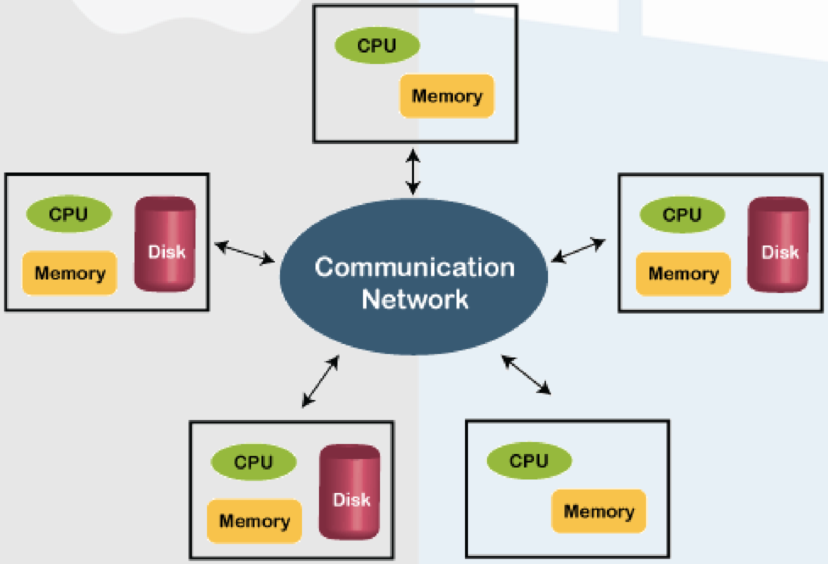
- These operating systems are designed to work on dedicated devices like automated teller machines (ATMs), airplane systems, digital home assistants, game consoles.

<u>Types of OS</u>

- **<u>Distributed Operating System</u>**

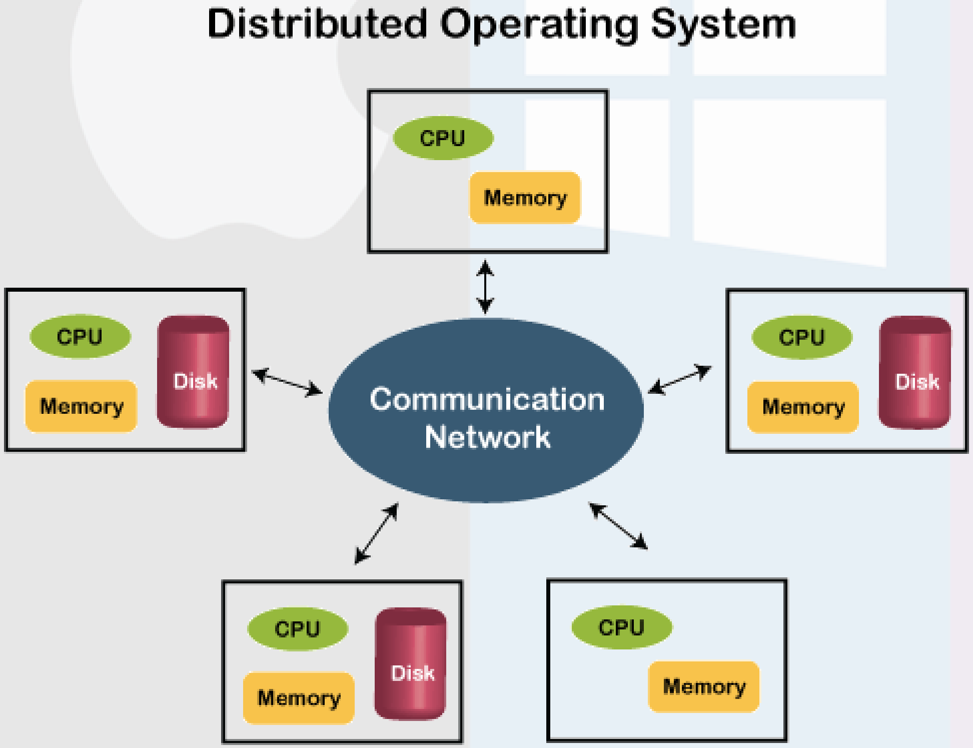
- Provides an environment in which multiple independent CPU or processor communicates with each other through physically separate computational nodes.

Distributed Operating System



Types of OS

- Each node contains specific software that communicates with the global aggregate operating system.



ROLES of OS’s

- There are 3 MAIN roles of any OS
  - Referee
  - Illusionist
  - Glue

ROLES of OS’s

- **Referee**
  - **Manage sharing of resources between different applications running on the same machine**
    - Stopping and Starting of programs
    - Isolation of applications from each other
    - Deciding which applications get resources when
    - Protect itself and other programs from malicious software

ROLES of OS’s

- **Illusionist**
  - **Provide clean, easy to use abstractions of physical resources**
    - Provides the illusion that a program has infinite memory
    - Provides the illusion that a program has the computers CPU all to itself
    - Allows programs to be written independently of the resources available
    - Since applications are written at a higher level of abstraction, the operating system can invisibly change the amount of resources allocated to each application

ROLES of OS's

- **Glue**
  - **Provide a set of common services that facilitate sharing among applications**
    - Cut/Paste works uniformly across the system
    - A file written by one application can be read by another
    - Common user interface routines which give applications a similar look and feel
    - Provides a layer separating applications from I/O devices

ROLES of OS’s

- **Referee**
  - **Manage sharing of resources, Protection**
    - Resource allocation, isolation, communication

- **Illusionist**
  - **Provide clean, easy to use abstractions of physical resources**
    - Infinite memory, dedicated machine
    - Higher level objects: files, users, messages
    - Masking limitations, virtualization

- **Glue**
  - **Common services**
    - Storage, Window system, Networking
    - Sharing, Authorization
    - Look and feel

So how does the OS perform these roles?

- Abstraction
- Concurrency
- Scheduling
- Persistence

Abstraction

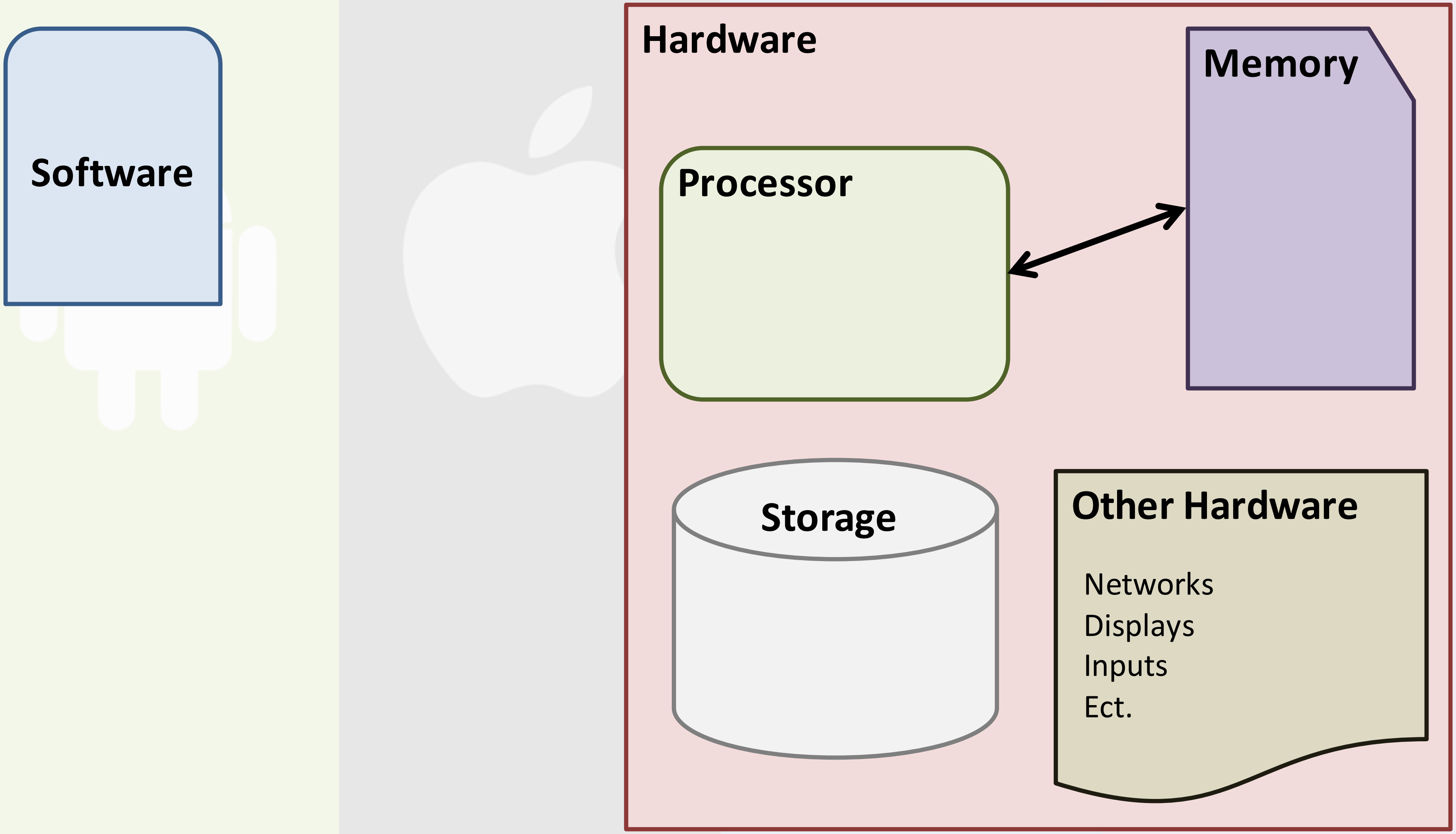
- Abstraction in general is a logical view of something. You don’t look at the (technical) details, but only at the principle.

Abstraction

- Abstraction in general is a logical view of something. You don’t look at the (technical) details, but only at the principle.

- **OS’s provide an abstract view of the computer HW. It doesn’t matter whether your sound card is from one manufacturer or another; the OS provides a unified way for applications to use it, they don’t need to know how does it work *internally*.**

Abstraction



Software

Hardware

Processor

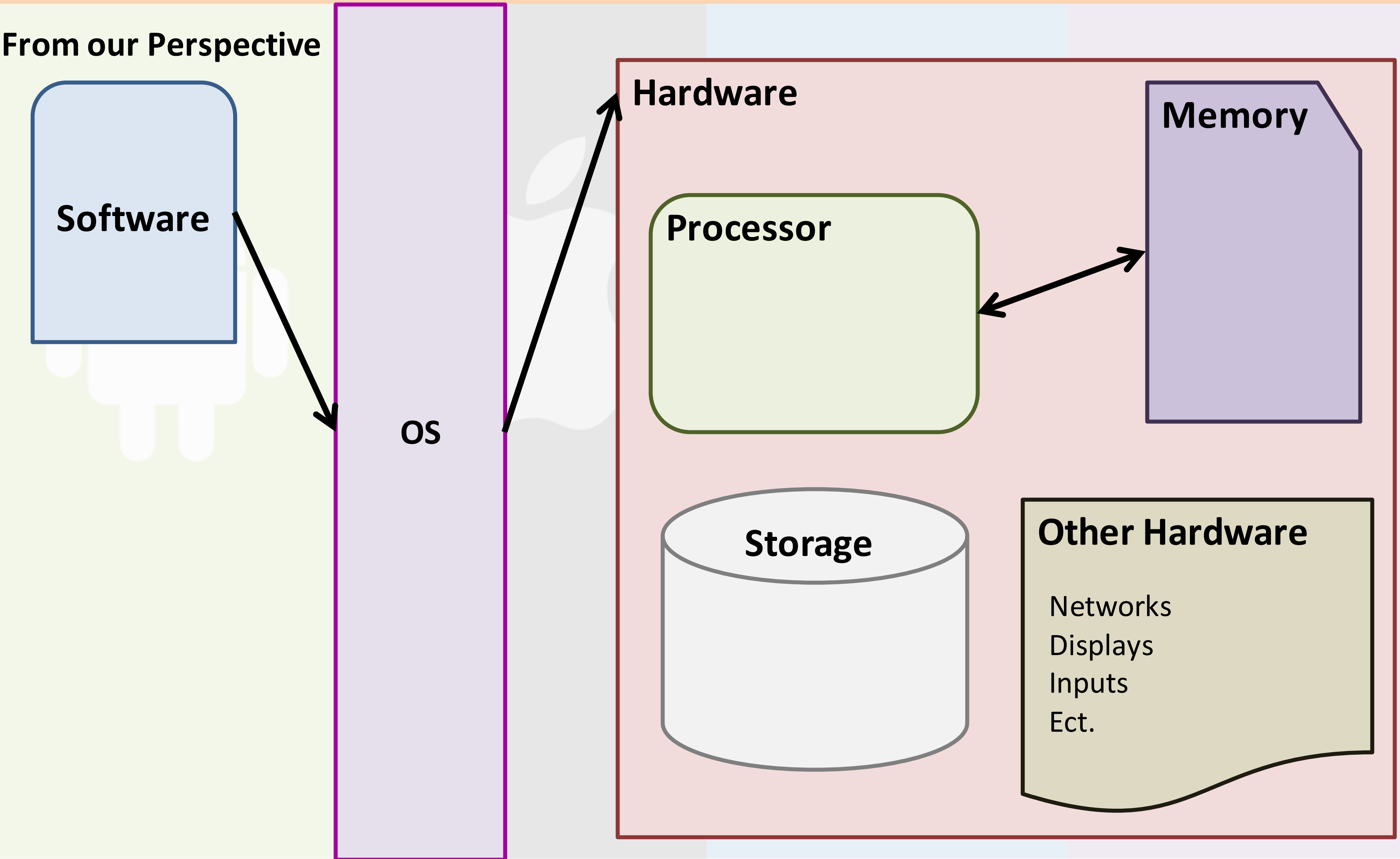
Memory

Storage

Other Hardware

Networks
Displays
Inputs
Ect.

Abstraction



From our Perspective

Software

OS

Hardware

Processor

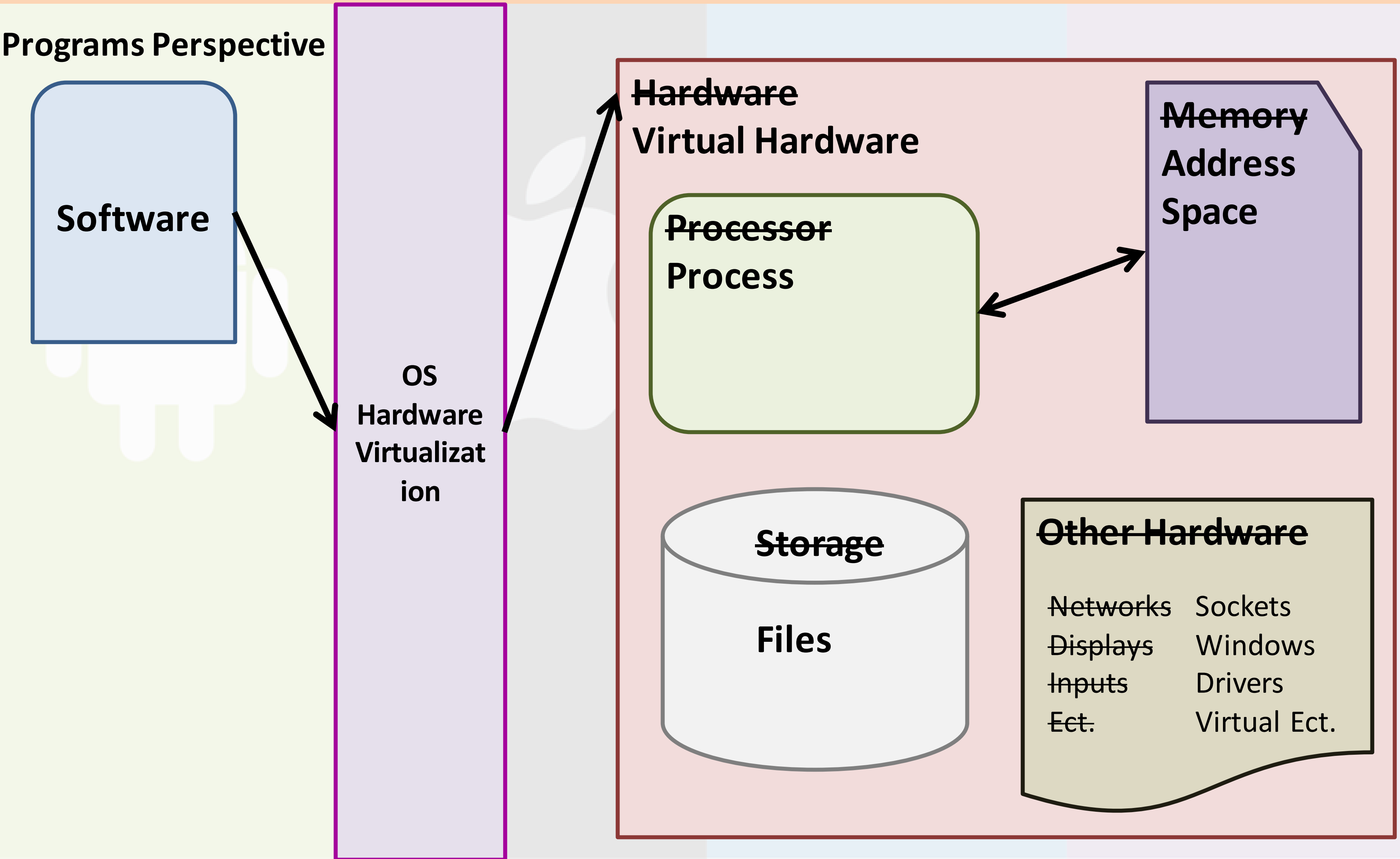
Memory

Storage

Other Hardware

Networks
Displays
Inputs
Ect.

Abstraction



Programs Perspective

Software

OS Hardware Virtualization

~~Hardware~~
Virtual Hardware

~~Processor~~
Process

~~Memory~~
Address Space

~~Storage~~
Files

~~Other Hardware~~

~~Networks~~	Sockets
~~Displays~~	Windows
~~Inputs~~	Drivers
~~Ect.~~	Virtual Ect.

Concurrency

- Concurrency is the tendency for things to happen at the same time in a system. Concurrency is a natural phenomenon, of course. In the real world, at any given time, many things are happening simultaneously.

Concurrency

- Concurrency is the tendency for things to happen at the same time in a system. Concurrency is a natural phenomenon, of course. In the real world, at any given time, many things are happening simultaneously.

- **When dealing with concurrency issues in software systems, there are generally two aspects that are important:**
    - being able to detect and respond to external events occurring in a random order,
    - ensuring that these events are responded to in some minimum required interval.

Concurrency

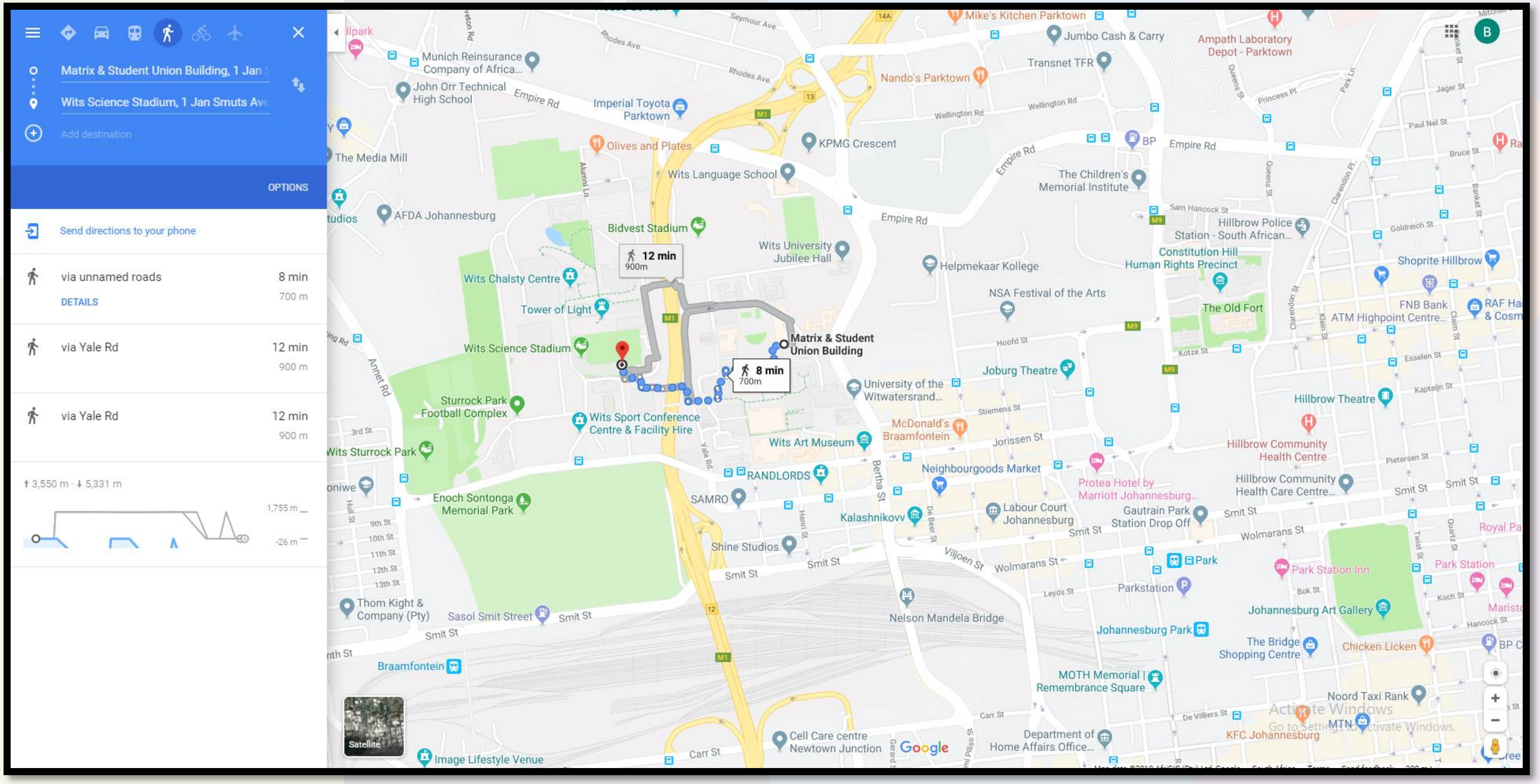
- Concurrency is the tendency for things to happen at the same time in a system. Concurrency is a natural phenomenon, of course. In the real world, at any given time, many things are happening simultaneously.

- When dealing with concurrency issues in software systems, there are generally two aspects that are important:
  - being able to detect and respond to external events occurring in a random order,
  - ensuring that these events are responded to in some minimum required interval.

- **If each concurrent activity evolved independently, in a truly parallel fashion, this would be relatively simple: we could simply create separate programs to deal with each activity.**
- **The challenges of designing concurrent systems arise mostly because of the interactions which happen between concurrent activities. When concurrent activities interact, some sort of coordination is required.**

Concurrency

- Google Maps



Matrix & Student Union Building, 1 Jan S

Wits Science Stadium, 1 Jan Smuts Ave

Add destination

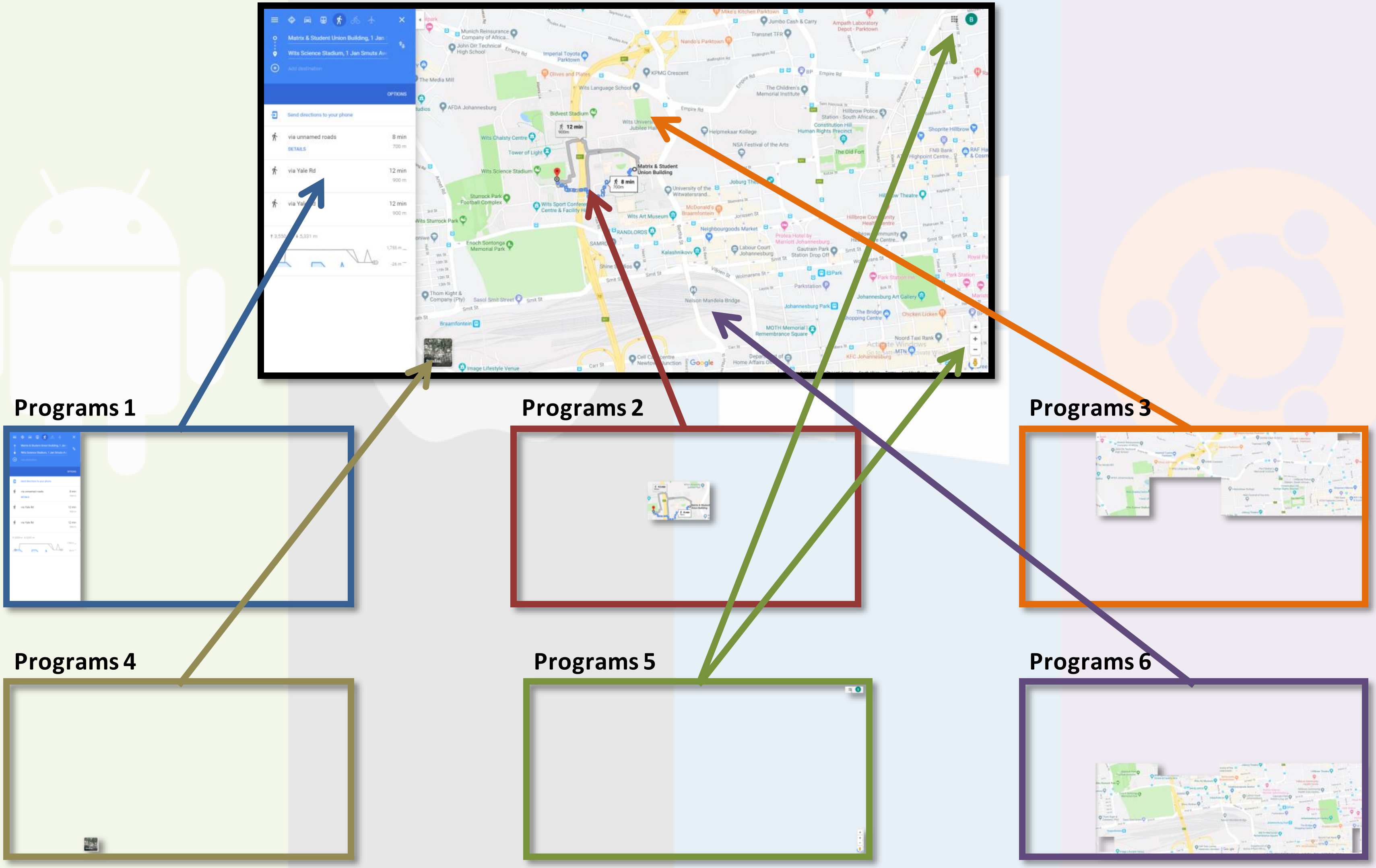
OPTIONS

Send directions to your phone

via unnamed roads DETAILS	8 min 700 m
via Yale Rd	12 min 900 m
via Yale Rd	12 min 900 m

↑ 3,550 m · ↓ 5,331 m

Concurrency



Matrix & Student Union Building, 1 Jan
Wits Science Stadium, 1 Jan Smuts Ave
OPTIONS
Send directions to your phone
via unnamed roads 8 min 700 m
DETAILS
via Yale Rd 12 min 900 m
via Yale Rd 12 min 900 m
12 min
8 min
Matrix & Student Union Building
Google

Programs 1

Programs 2

Programs 3

Programs 4

Programs 5

Programs 6

Scheduling

- **In computing, scheduling is the method by which work is assigned to resources that complete the work.**

- The work may be virtual computation elements such as threads, processes or data flows, which are in turn scheduled onto hardware resources such as processors, network links or expansion cards.

Scheduling

- In computing, scheduling is the method by which work is assigned to resources that complete the work.

- **The work may be virtual computation elements such as threads, processes or data flows, which are in turn scheduled onto hardware resources such as processors, network links or expansion cards.**

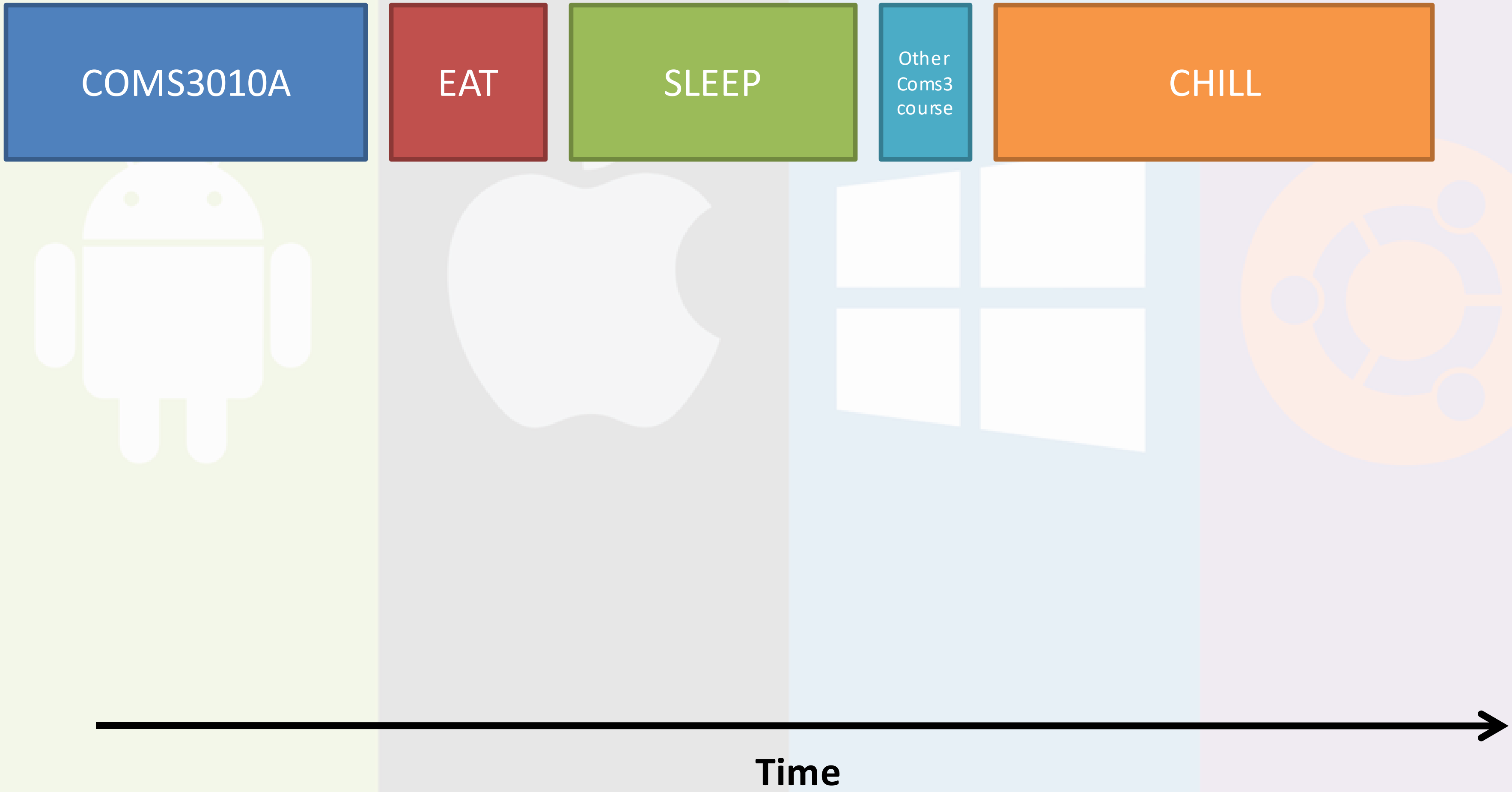
Scheduling

- In computing, scheduling is the method by which work is assigned to resources that complete the work.

- The work may be virtual computation elements such as threads, processes or data flows, which are in turn scheduled onto hardware resources such as processors, network links or expansion cards.

- **A scheduler is what carries out the scheduling activity.**
- **Schedulers are often implemented so they keep all computer resources busy, allow multiple users to share system resources effectively, or to achieve a target quality of service.**

Scheduling



COMS3010A

EAT

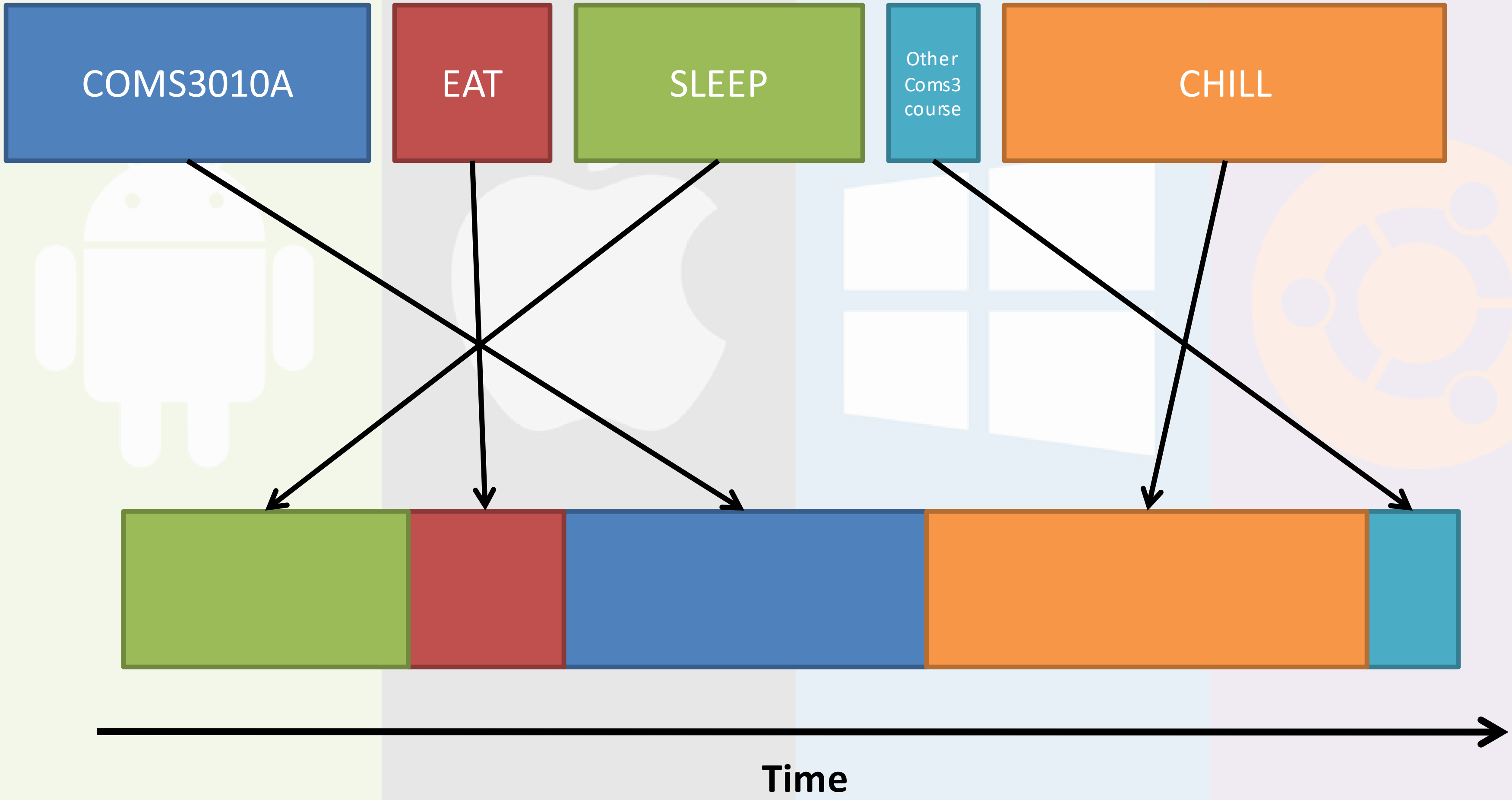
SLEEP

Other Coms3 course

CHILL

Time

Scheduling



Persistence

- In computer science, persistence refers to the characteristic of state that outlives the process that created it.

Persistence

- In computer science, persistence refers to the characteristic of state that outlives the process that created it.

- **This is achieved in practice by storing the state as data in computer data storage. Programs have to transfer data to and from storage devices and have to provide mappings from the native programming-language data structures to the storage device data structures.**

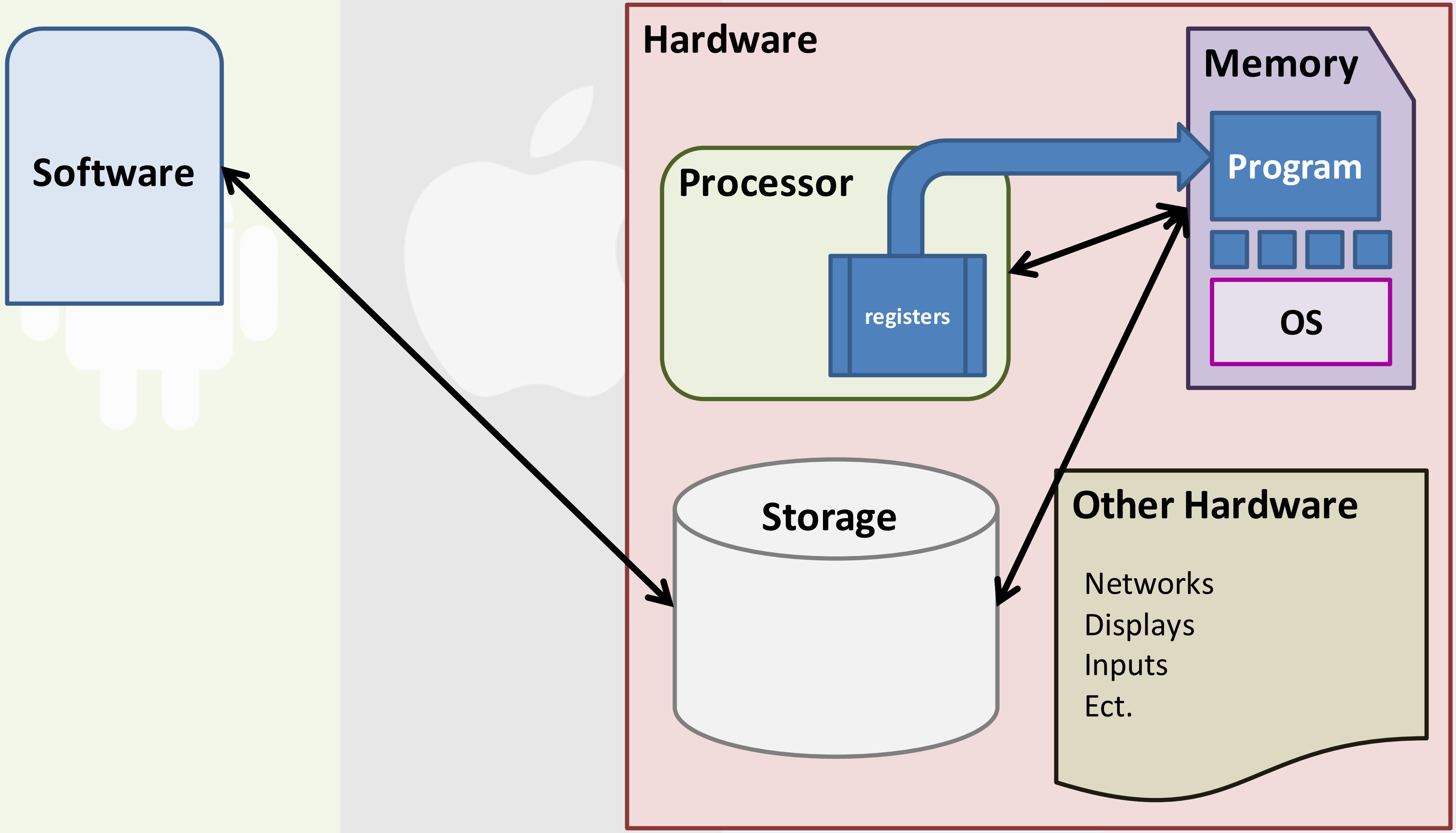
Persistence

- In computer science, persistence refers to the characteristic of state that outlives the process that created it.

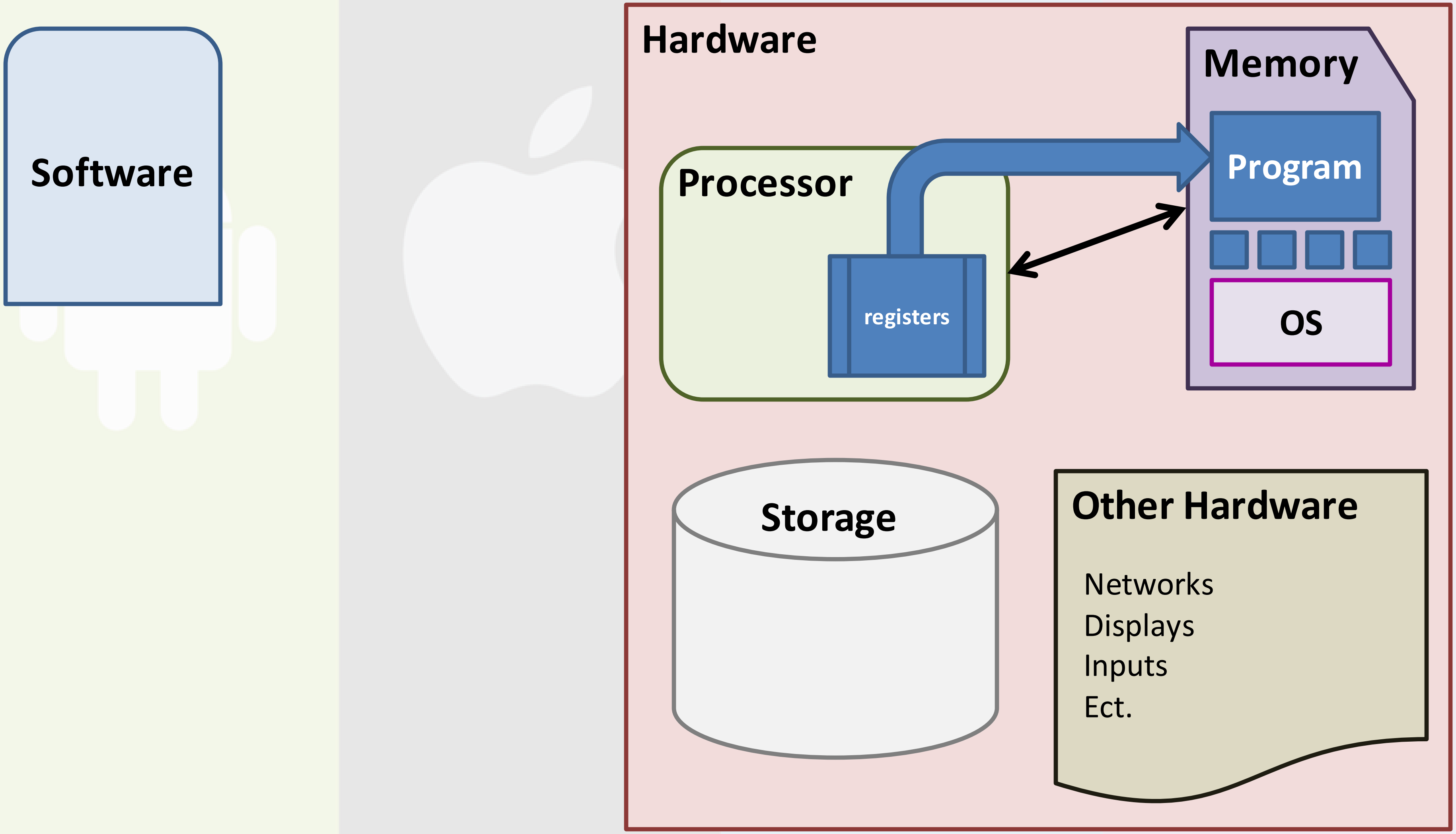
- This is achieved in practice by storing the state as data in computer data storage. Programs have to transfer data to and from storage devices and have to provide mappings from the native programming-language data structures to the storage device data structures.

- **Picture editing programs or word processors, for example, achieve state persistence by saving their documents to files.**

Persistence



OS Basics : Loading



Software

Hardware

Memory

Program

OS

Processor

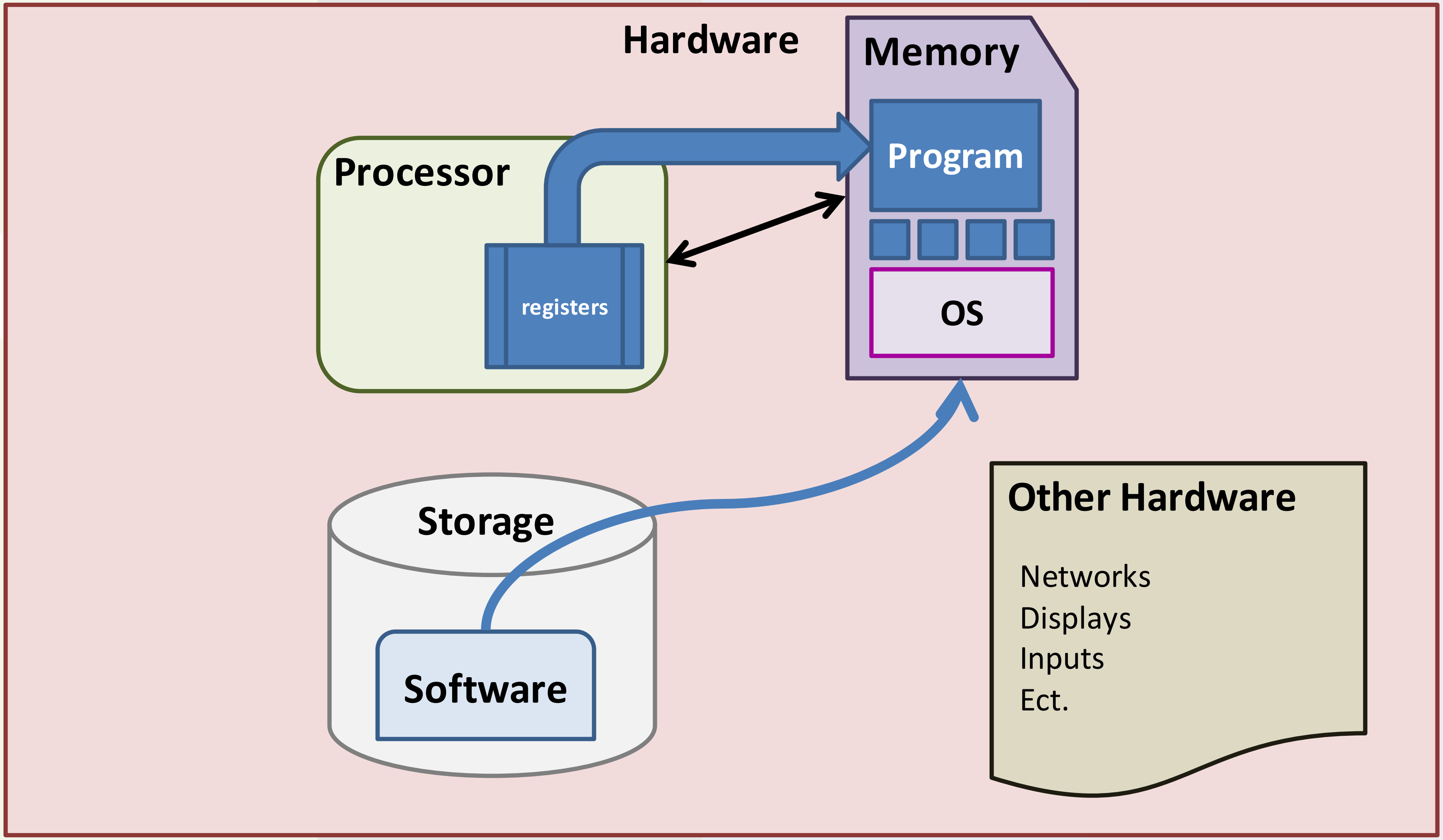
registers

Storage

Other Hardware

Networks
Displays
Inputs
Ect.

OS Basics : Loading



History

- In the Beginning

History

- The first operating system was created by General Motors in 1956 to run a single IBM mainframe computer.

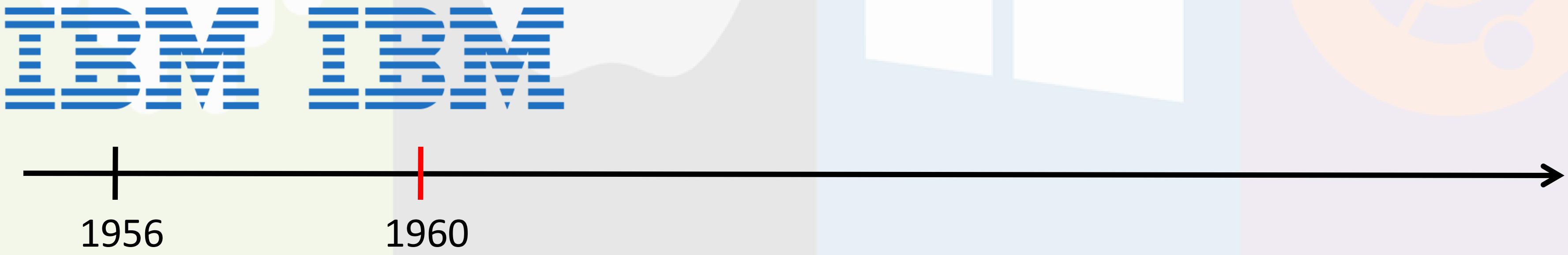
1956

```
C:\
65080201 bytes used in 61 items.
X DISCS                      10-10-
X FOURF                      10-10-
X HARD_DRV                   10-10-
X [illegible]                10-10-
X MACROS                     10-10-
X MEMORY~1                   10-10-
X PATCHES                    10-10-
X PROFILES                   10-10-
X SCREEN~1                   10-10-
X SHORTC~1                   10-10-
X ST_GAMES                   10-10-
X TOSIMA~1                   10-10-
X ZIPS                       10-10-
  AUTO      STS   790274     10-10-
  AUTO_R~1  STS   790274     10-10-
```

```
C:\FOURF\
used in 23 items.
DAT      3960  10-
DAT      3960  10-
MBK     13074  10-
DAT      3960  10-
INF        54  10-
PRG    173982  10-
MBK      5986  10-
DAT      3960  10-
MBK      3602  10-
DAT      3960  10-
DAT      3960  10-
```

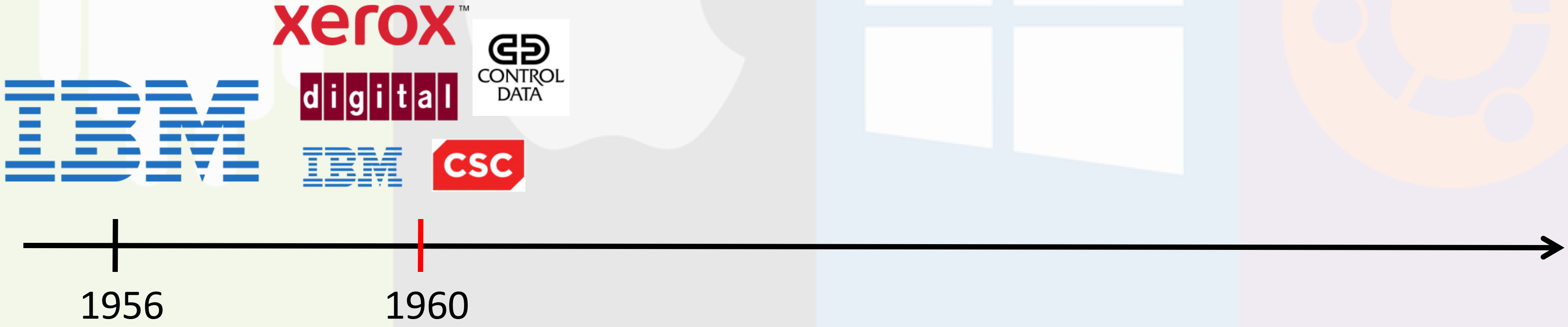
History

- In the 1960s, IBM was the first computer manufacturer to take on the task of operating system development and began distributing operating systems with their computers

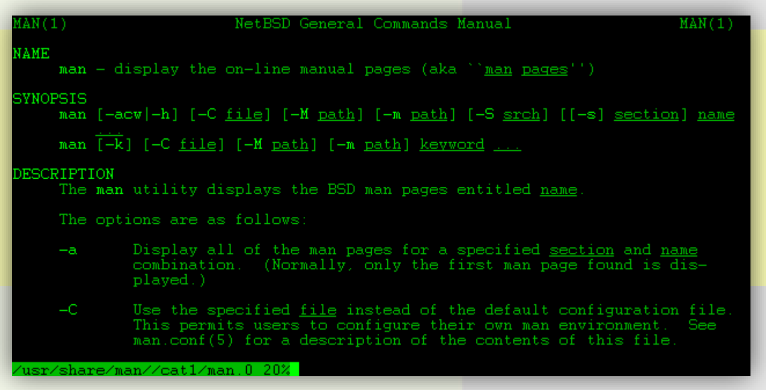


History

- However, IBM wasn't the only vendor creating operating systems during this time. Control Data Corporation, Computer Sciences Corporation, Burroughs Corporation, GE, Digital Equipment Corporation, and Xerox all released mainframe operating systems in the 1960s as well

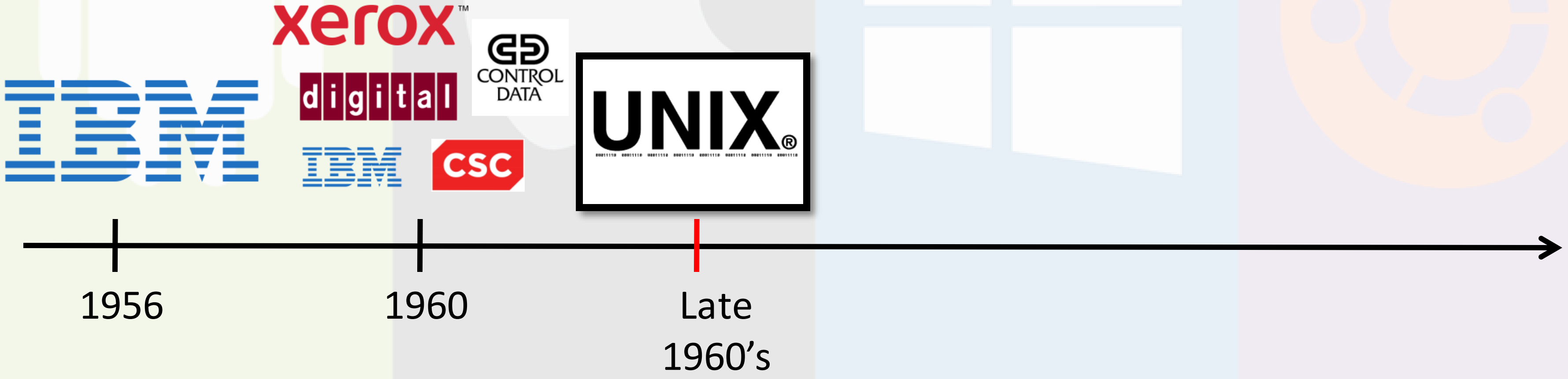


History



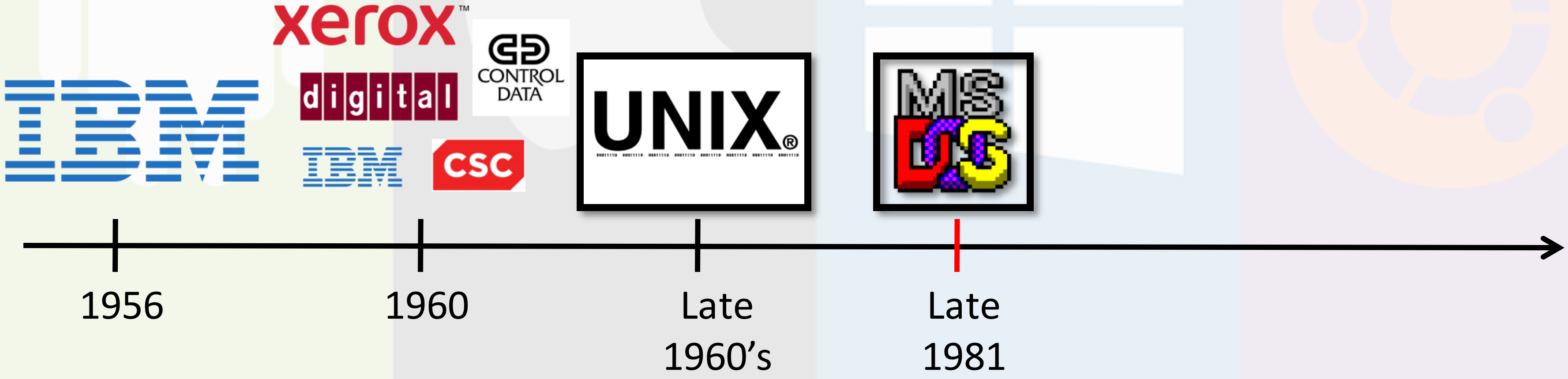
```
MAN(1)                  NetBSD General Commands Manual                  MAN(1)

NAME
     man - display the on-line manual pages (aka ``man pages'')

SYNOPSIS
     man [-acw|-h] [-C file] [-M path] [-m path] [-S srch] [[-s] section] name
         ...
     man [-k] [-C file] [-M path] [-m path] keyword ...

DESCRIPTION
     The man utility displays the BSD man pages entitled name.

     The options are as follows:

     -a      Display all of the man pages for a specified section and name
             combination.  (Normally, only the first man page found is dis-
             played.)

     -C      Use the specified file instead of the default configuration file.
             This permits users to configure their own man environment.  See
             man.conf(5) for a description of the contents of this file.

/usr/share/man//cat1/man.0 20%
```

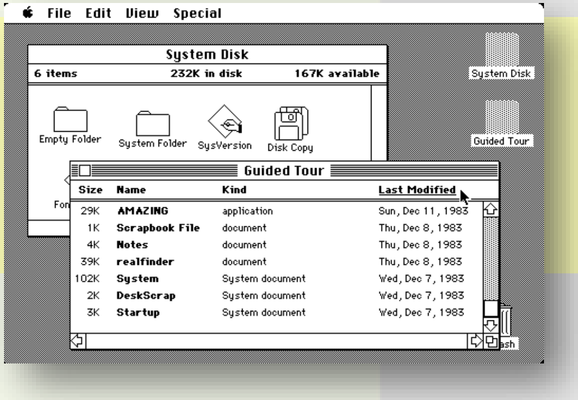
- In the late 1960s, the first version of the Unix operating system was developed. Written in C, and freely available during it’s earliest years, Unix was easily ported to new systems and rapidly achieved broad acceptance.



History

- The first OS built by Microsoft wasn't called Windows, it was called MS-DOS and was built in 1981 by purchasing the 86-DOS operating system from Seattle Computer Products and modifying it to meet IBM's requirements.





System Disk

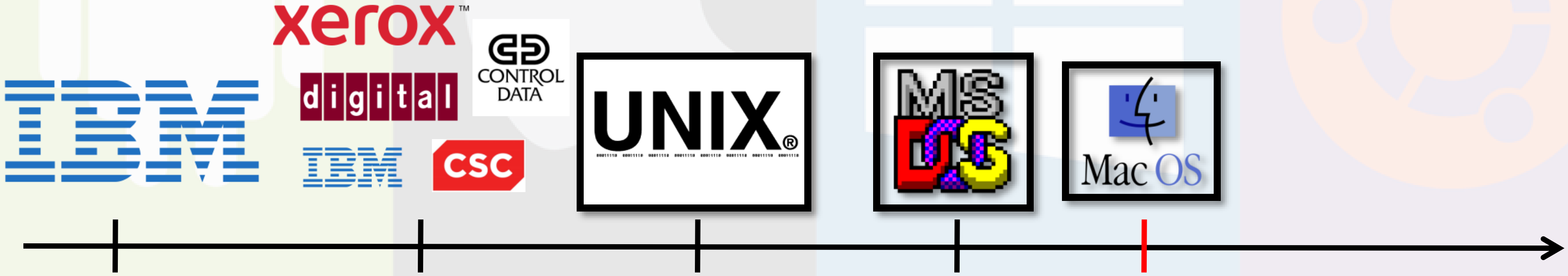
6 items 232K in disk 167K available

Guided Tour

Size	Name	Kind	Last Modified
29K	AMAZING	application	Sun, Dec 11, 1983
1K	Scrapbook File	document	Thu, Dec 8, 1983
4K	Notes	document	Thu, Dec 8, 1983
39K	realfinder	document	Thu, Dec 8, 1983
102K	System	System document	Wed, Dec 7, 1983
2K	DeskScrap	System document	Wed, Dec 7, 1983
3K	Startup	System document	Wed, Dec 7, 1983

History

- Apple released the original Macintosh on January 24, 1984. This was the first version of the system software which would later go on to be called MacOS



1956

1960

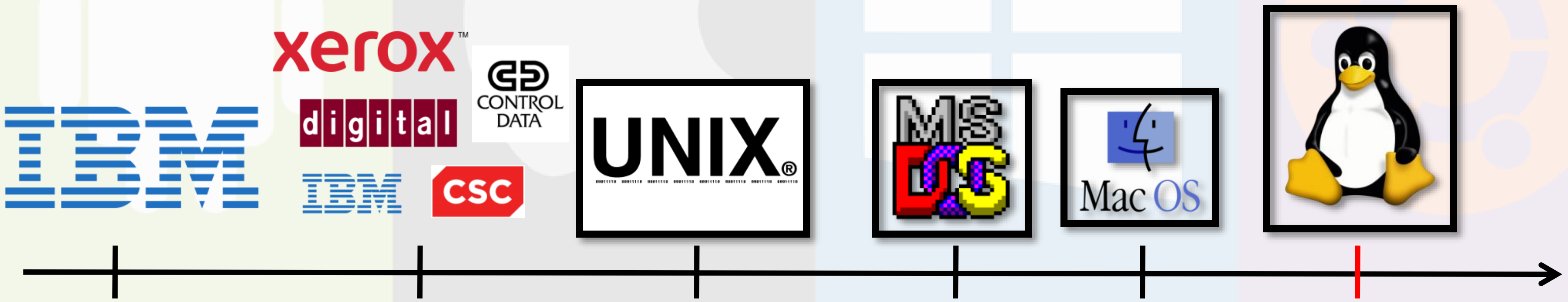
Late 1960’s

Late 1981

1984

History

- Linux was first released on September 17, 1991, by Linus Torvalds.



1956

1960

Late 1960’s

Late 1981

1984

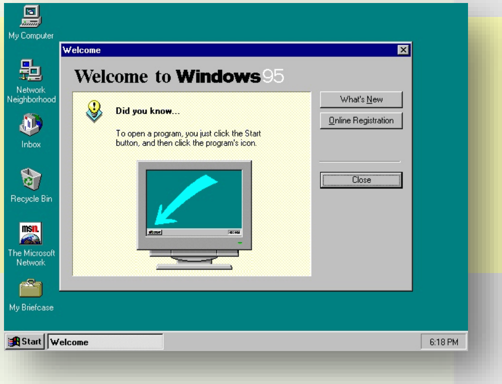
Late 1991

History

- Developed by Microsoft as a high-end server Operating System, the NT code became the basis for Operating Systems to this day.

MICROSOFT
WINDOWSNT

1993



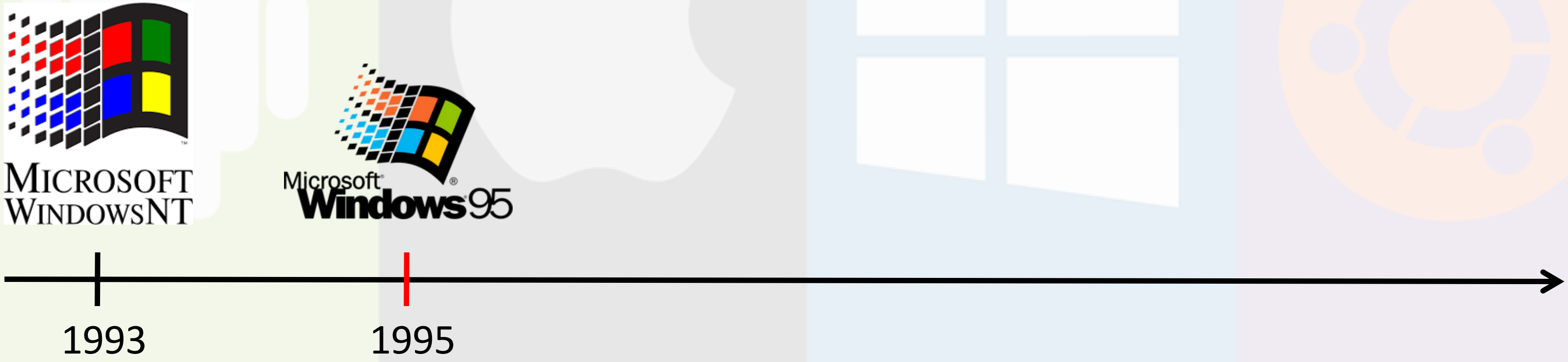
Welcome to Windows95

Did you know...

To open a program, you just click the Start button, and then click the program's icon.

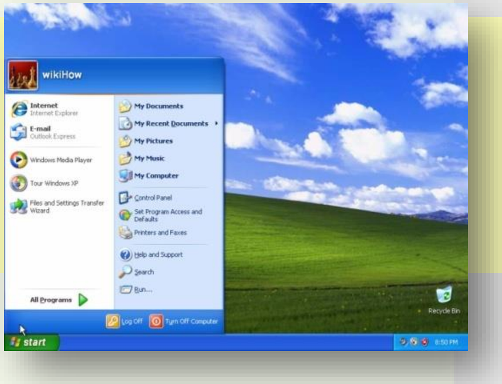
History

- Developed by Microsoft, it was the first Microsoft Operating system to have a graphical user interface built into it



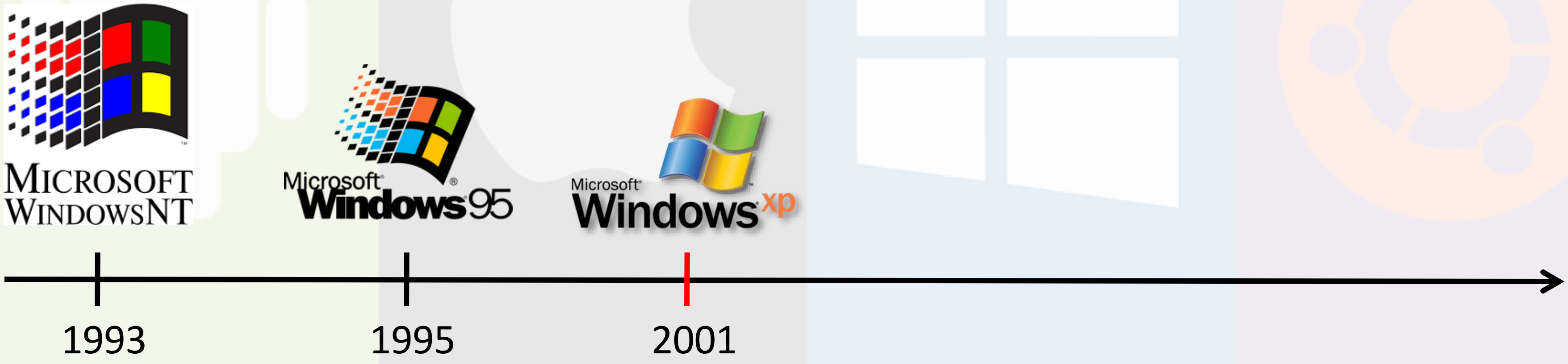
1993

1995



History

- XP was an enhanced version of Windows 2000 code base. XP became widely popular and is used extensively today



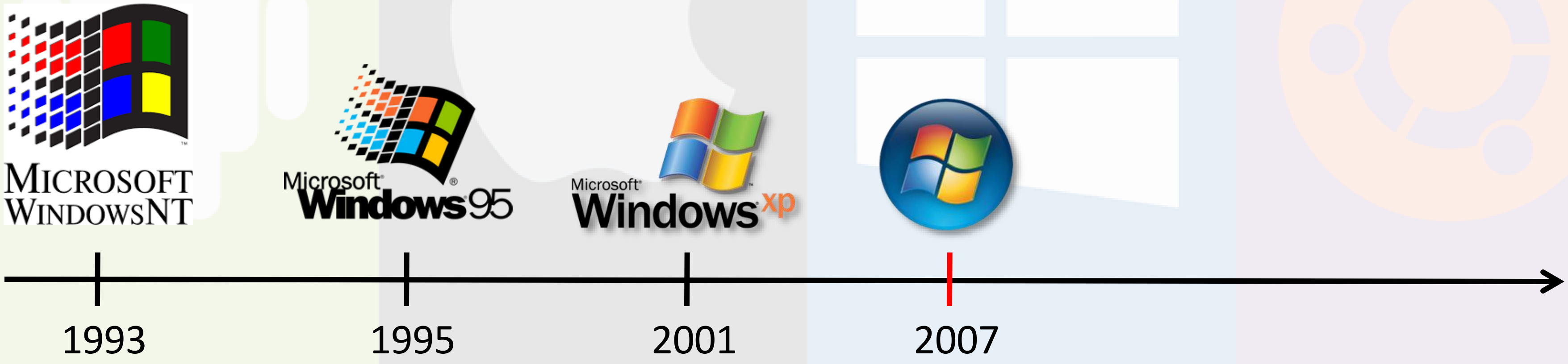
1993

1995

2001

History

- Vista had been slow in taking off.



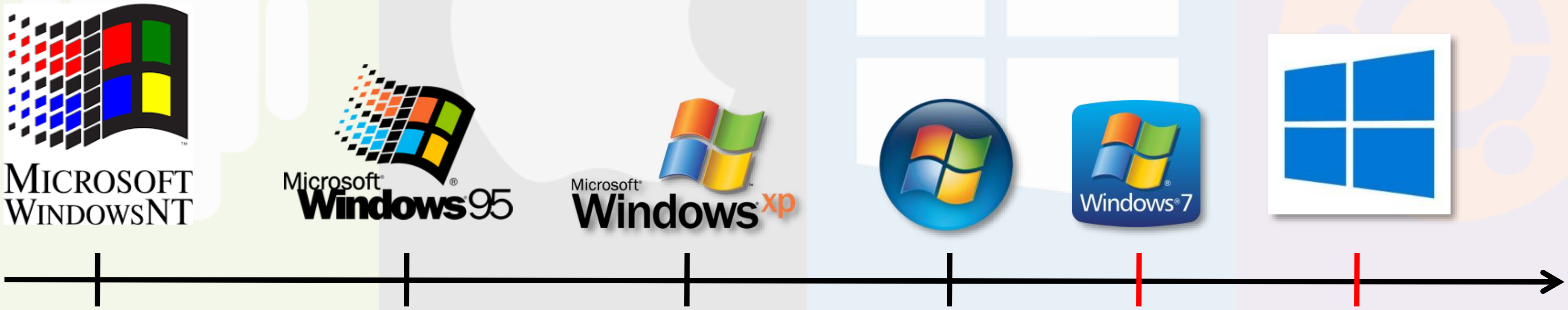
1993

1995

2001

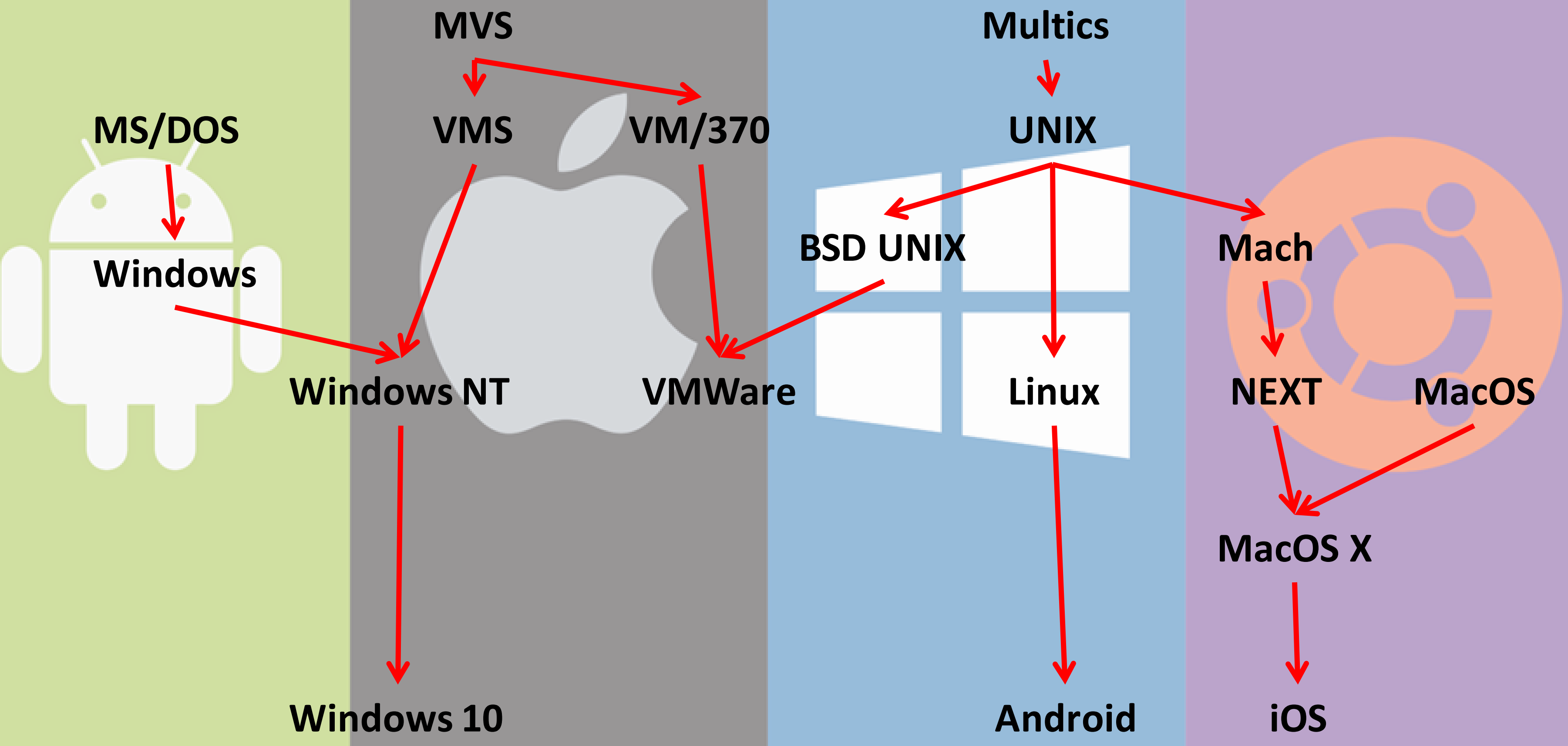
2007

History



1993 1995 2001 2007 2009 2014

Where did we come from? Where did we go?



Genealogy of Operating Systems