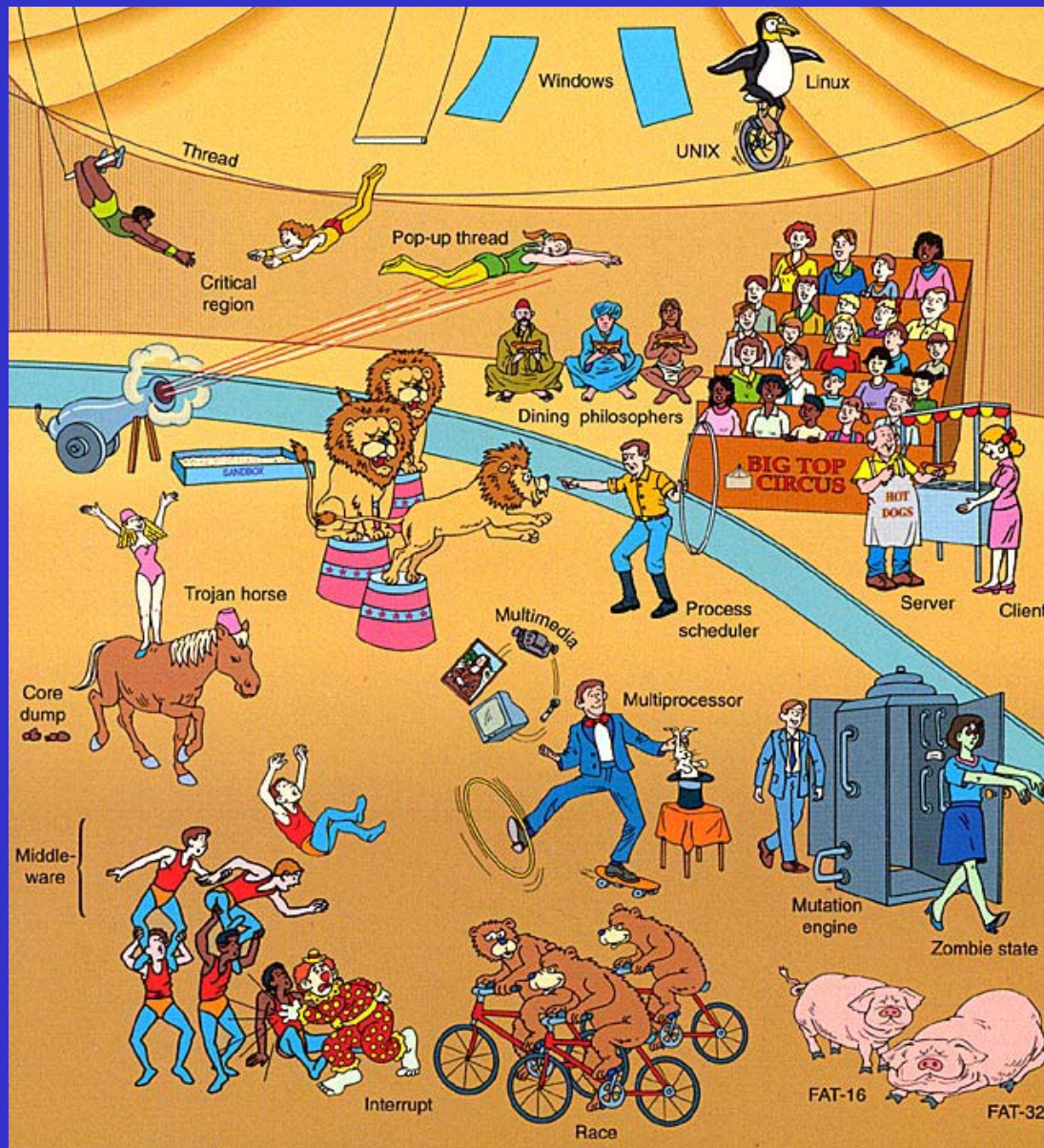


Introduction to the UNIX environment

Enrique Blanco



Cover from the book Modern operating systems. A. Tanenbaum. Prentice Hall

First generation (mid40s-50s):

- Tubes, switches and light displays, no operating system, human operators

Second generation (mid50s-60s):

- Transistors, batch processing, monitors, jobs, cards

Third generation (mid60s-80s):

- Chips (LSI, VLSI), time sharing systems, transactions, tapes, drives

Our generation:

- Microprocessors, distributed systems, parallel processing, internet, ...

We have moved away from the hardware forever !!!

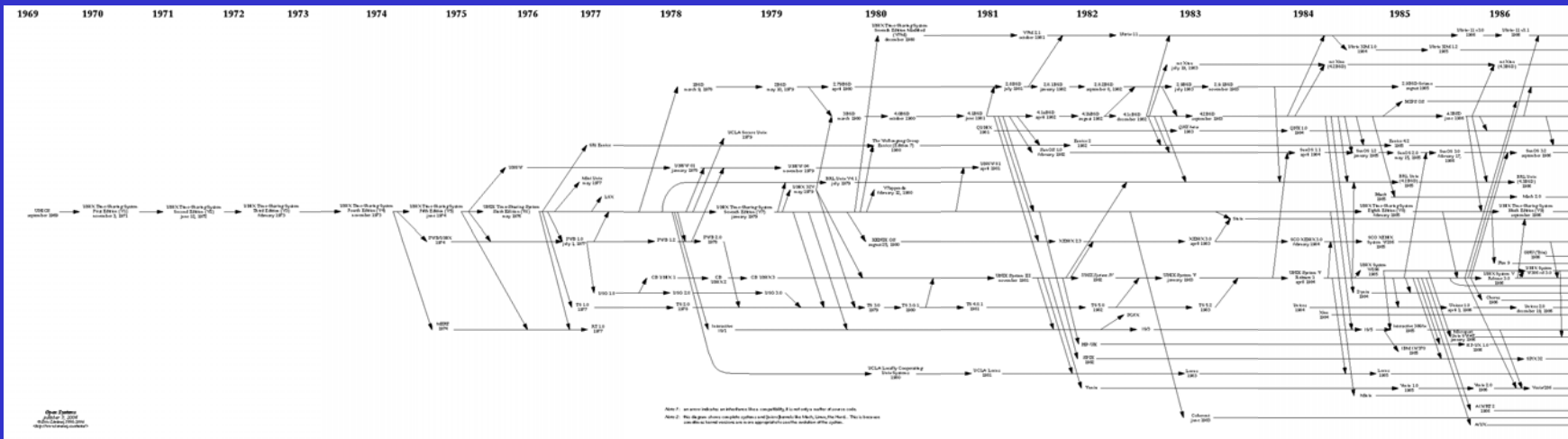
1969: UNIX (from MULTICS) by

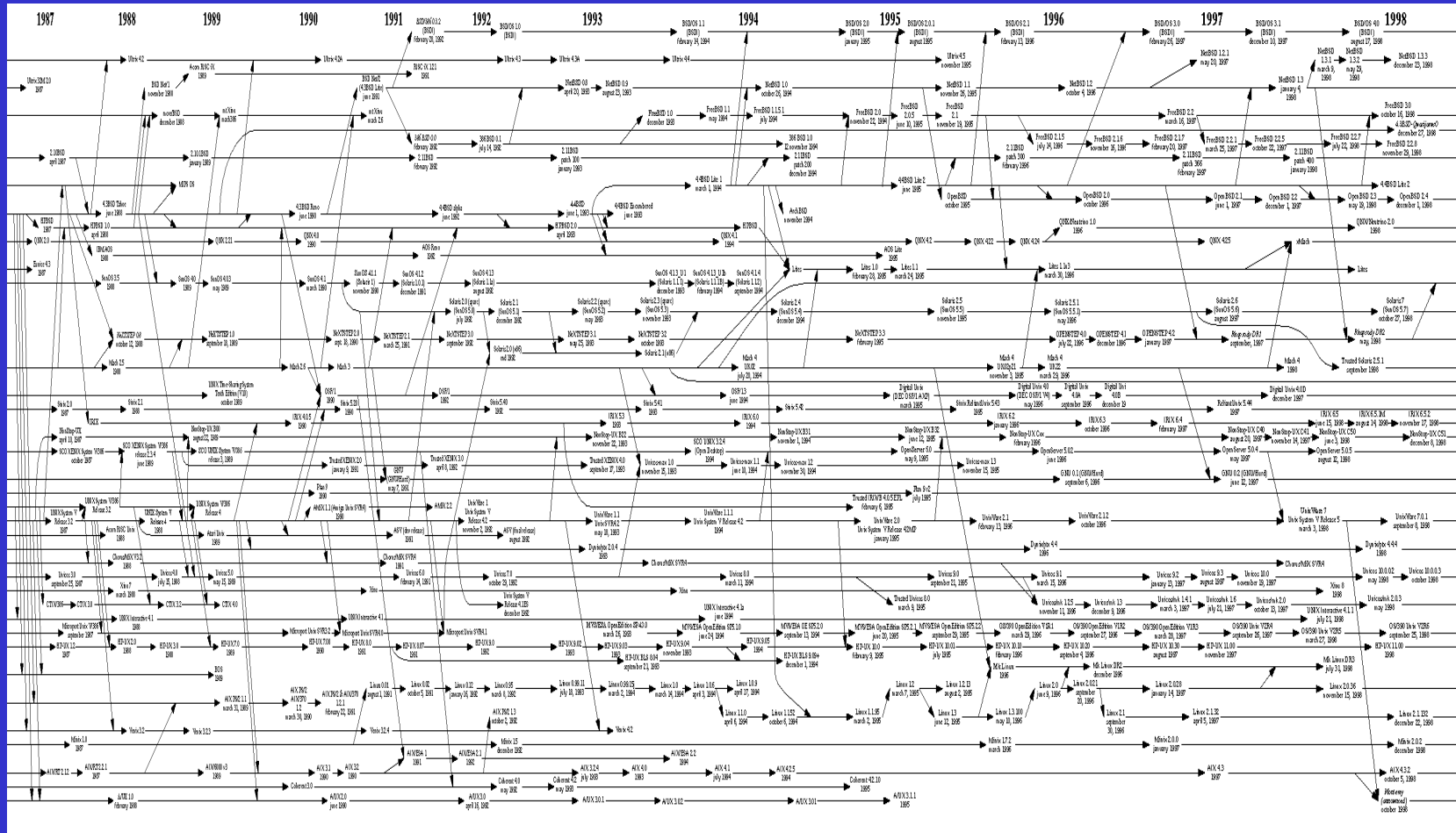


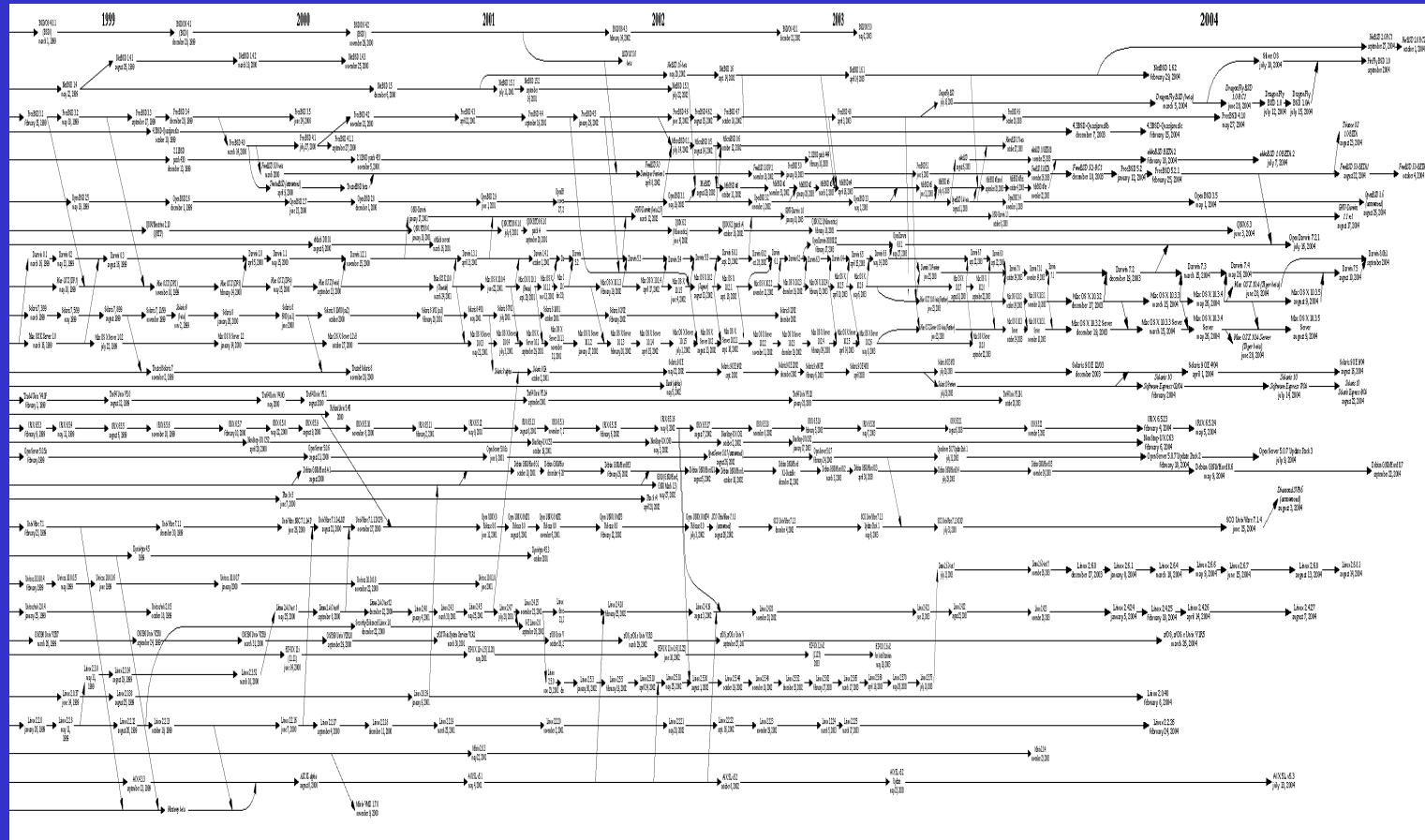
Ken Thompson



Denis Ritchie







<http://www.levenez.com/unix/>

The OS is the interface user-machine that provides the management of different resources:

- Programs and processes (processors)
- Memory (and other data storage elements)
- In/Out devices (mouse, keyboard, printer, ...)
- Files and directories
- Protection and safety
- Accounting information
- Network

The OS is a **Virtual Machine** that substitutes the real hardware

The objective is to create the mirage of a machine with unlimited resources:

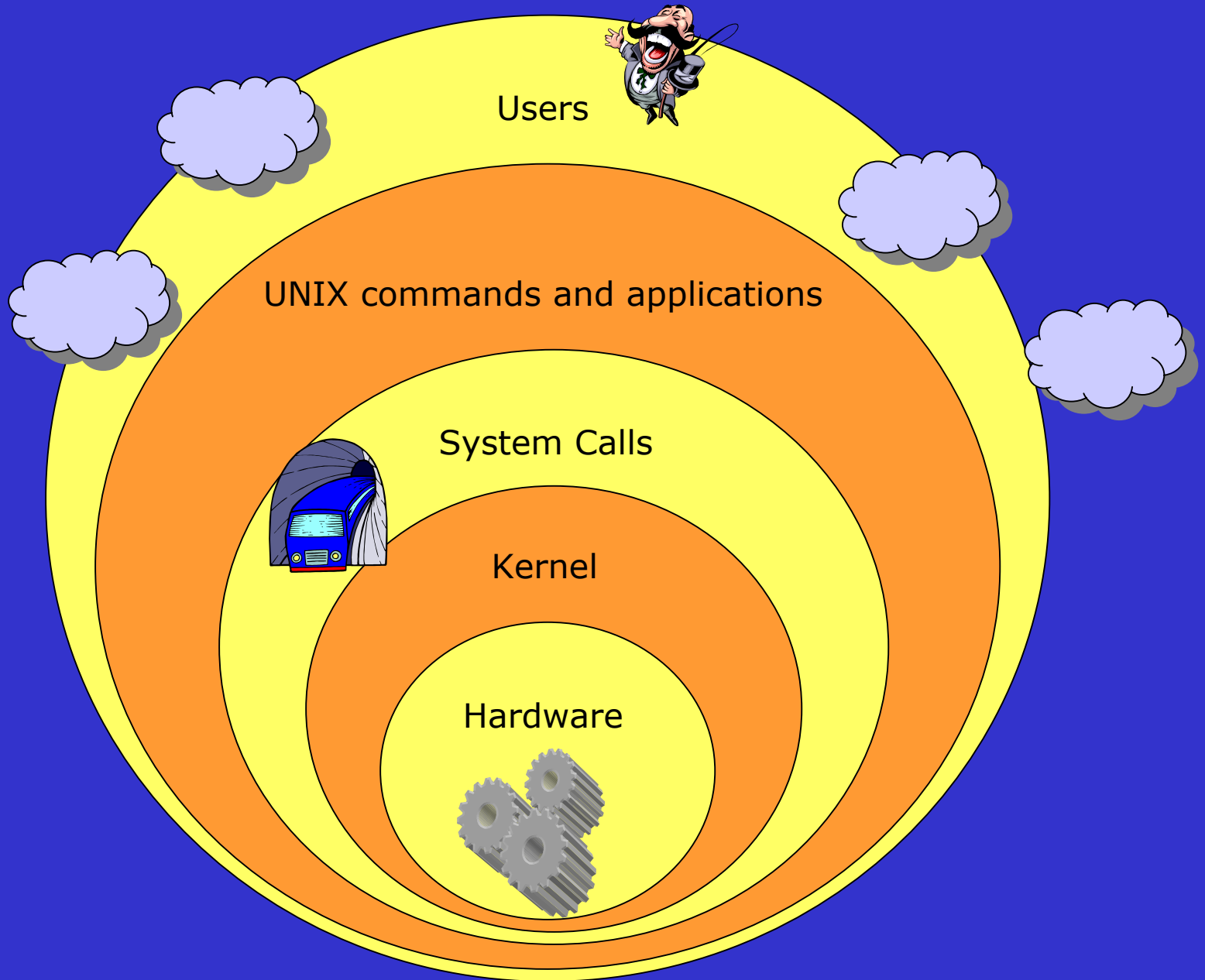
- One available processor for running every user program
- Infinite amount of available physical memory
- The physical devices can be accessed uniformly

VIRTUAL Vs PHYSICAL RESOURCES

The OS is in charge of masking the complexity of every hardware element, providing a consistent interface to devices and services for users and programs

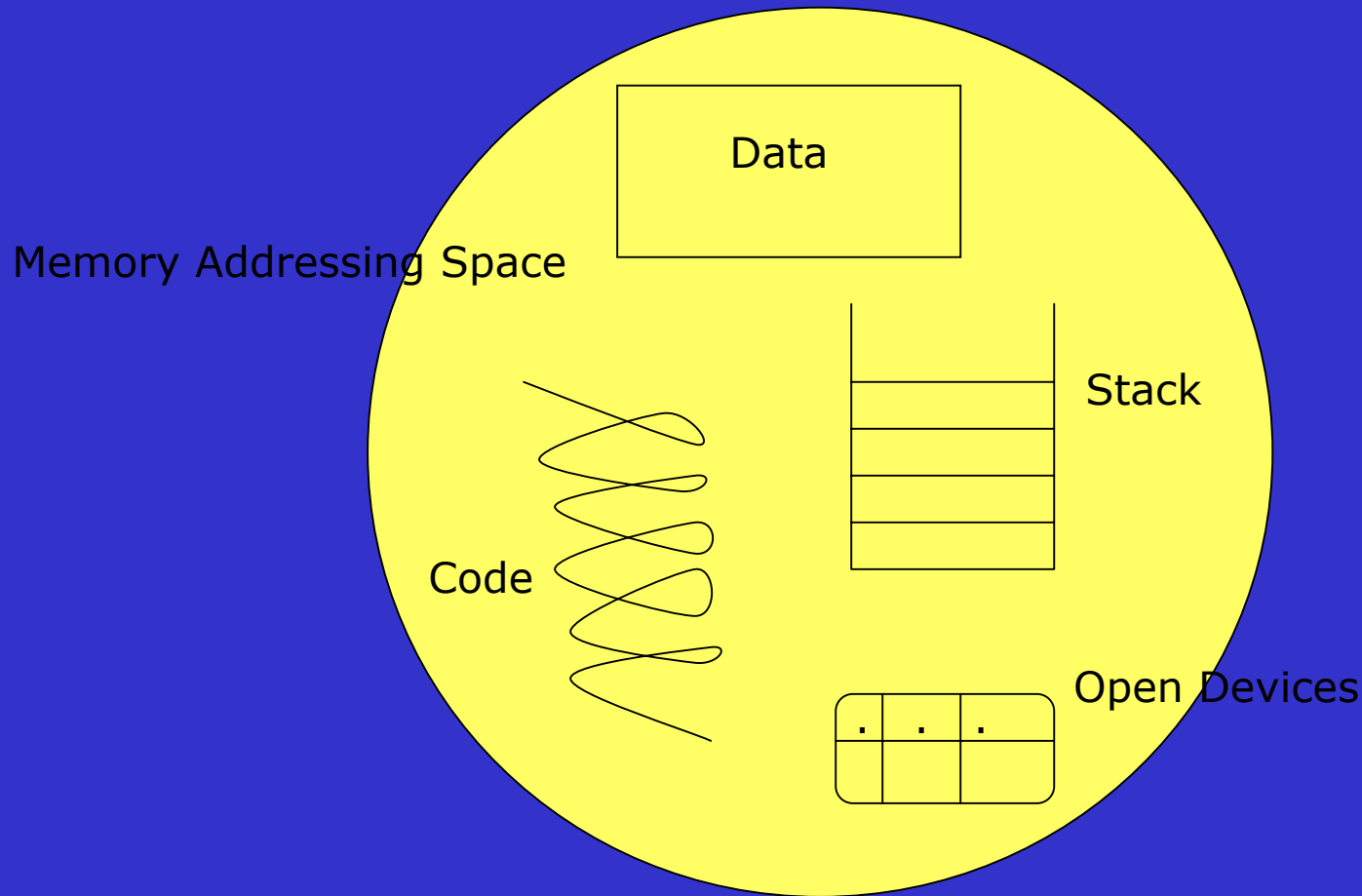
Independence = Isolation from the computer (High level)

Classical OS hierarchical structure (layers)



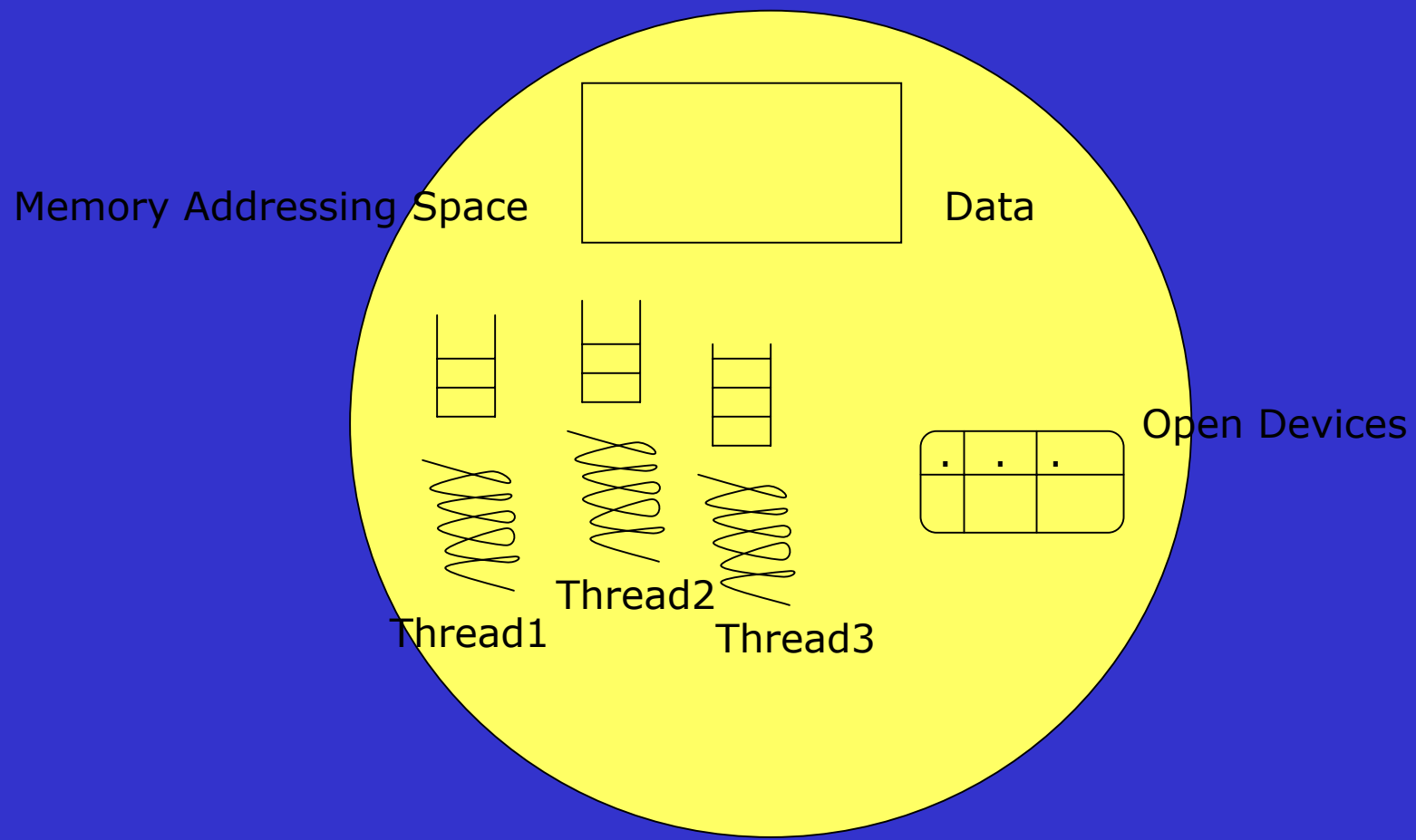
What is a process?

A process/task is a running environment that provides the resources to execute a particular instance of a program:



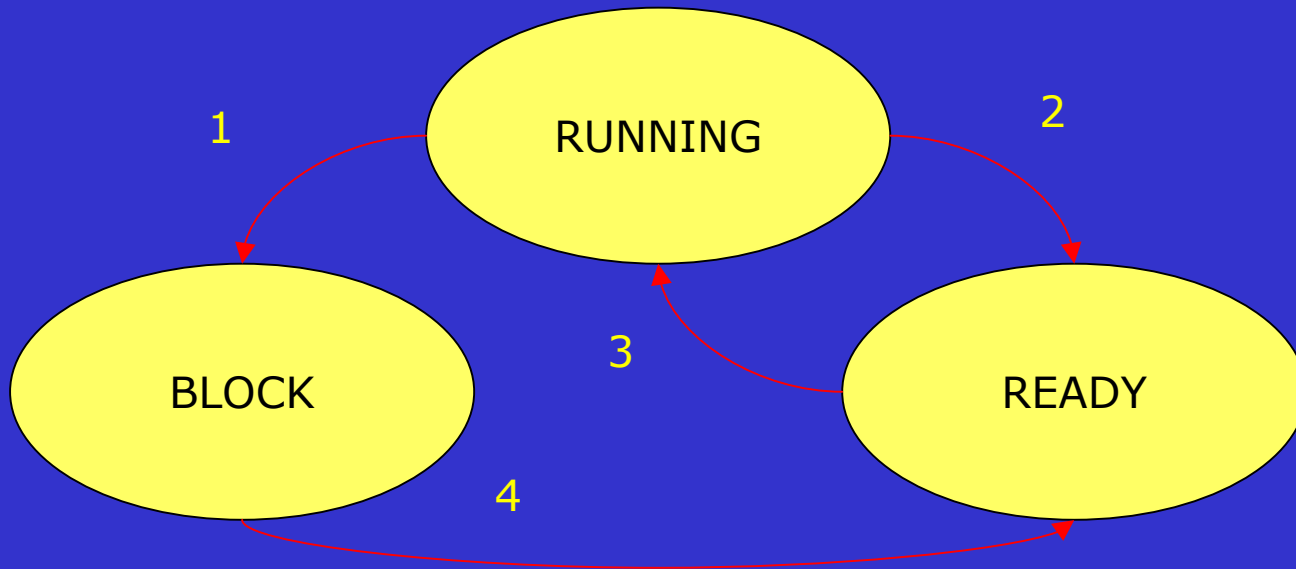
The OS schedules the processes (distribute the processor running time among them)

Several parts of the program can be executed in parallel in different virtual processors



- Edition: source code in a High-level language (C, Java, ...)
 - Variables and functions
[program.c]
- Compilation: syntax and semantic correctness
 - Table of symbols and list of unsolved references
[program.o]
- Linking: translate the symbols into numbers (memory references)
 - Generate the binary file adjusting the memory addresses
[program]
- Load: transport the binary file to physical memory
 - Create the process and assign resources (memory,...)

Interpreter: compilation, linking and loading line by line (no generation of additional files)



1. Waiting for I/O
2. Another process is scheduled
3. This process is scheduled
4. Received I/O data

1. Waiting for user commands from terminal
2. Receive a user request (e.g a UNIX command)
3. Create a process (child) and assign resources to execute the request

- 4A. Foreground (**synchronous**):
 - Waiting for the end of the child to continue...
 - Go to Step1

- 4B. Background (**asynchronous**):
 - Go to Step1 (child works simultaneously)

- Device driver:
 - interface between the physical device and other elements
- Physical devices are slow in comparison to the processor
- Sometimes, devices can be shared among many users

UNIX Device Independence:

- Mask the physical complexity of devices and provide common procedures to interact with them
- It must be possible to design programs that process files from any source
- In UNIX, the physical devices are integrated as normal files
- I/O techniques: Spooling and Buffering

File: collection of bits with a name saved in a physical storage device

File system: mechanism to access the file using its symbolic name

Directory: data structure that groups a set of files

File operations: open, close, read, write, append, ...

UNIX file management is implemented using indexed-nodes (i-nodes):

Device

Number of links

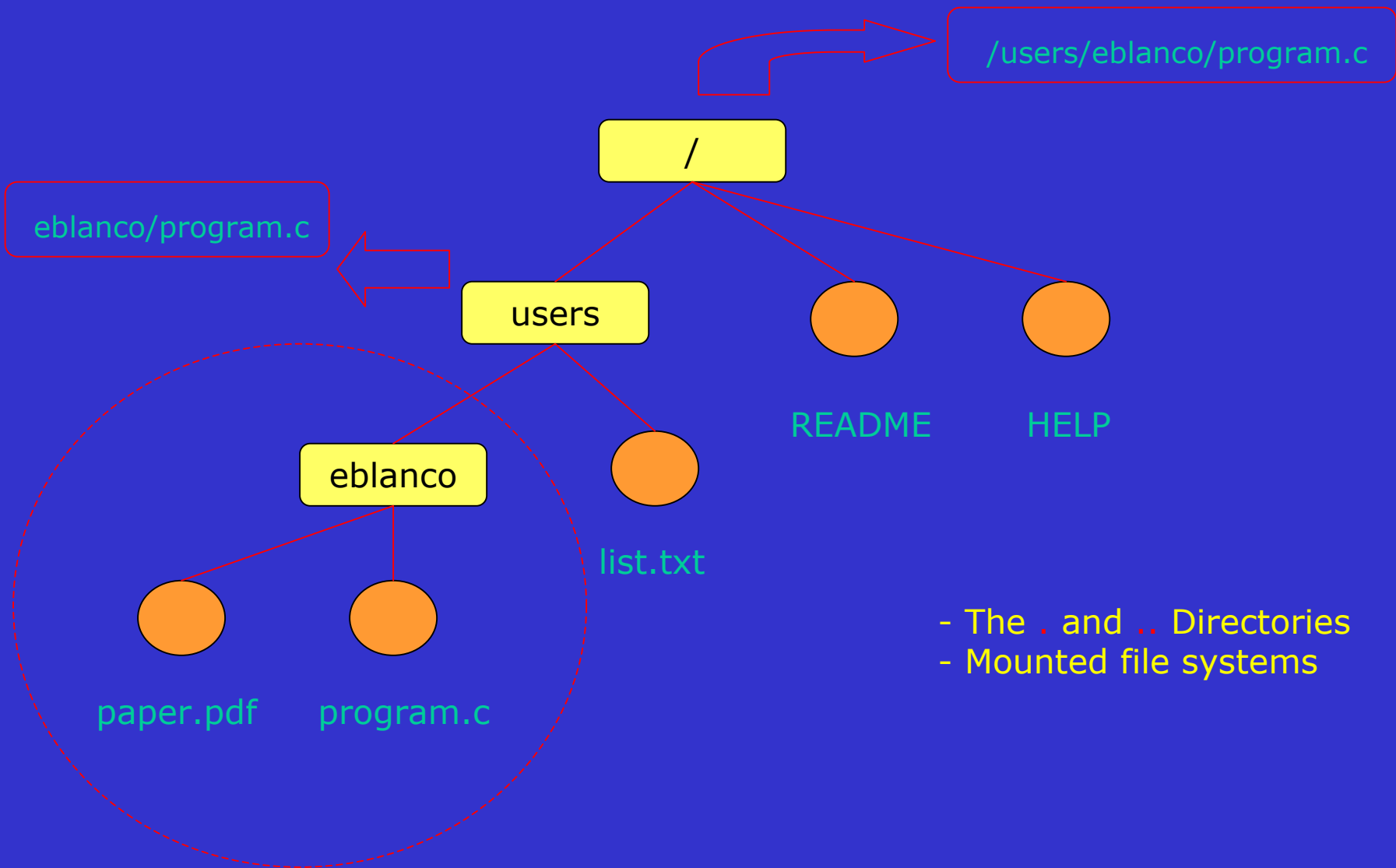
Owner

Size

Data creation, accession, modification

Pointers to several data blocks

Absolute and relative paths



- The `.` and `..` Directories
- Mounted file systems

- Hard link:
 - Several names in different paths for the same file (same i-node)
 - Any modification using one of the names will affect the others
- Soft link:
 - One file contains the reference to another one (path)
 - Each file have its own i-node

- 3 operations [RWX]:

Read	Write	eXecute
------	-------	---------

- 3 domains:

User	Group	Others
------	-------	--------

r	w	-	r	-	-	r	-	-
1	1	0	1	0	0	1	0	0
	6			4			4	
	u			g			o	