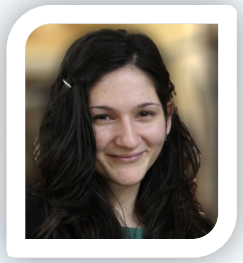
SISTEMI OPERATIVI

Prof. G. Ianni



Dott. D. Angilica

Sito del corso:

qui

Gruppo FB («Sistemi Operativi e Reti – UNICAL - INFORMATICA»):

https://bit.ly/2Er8HBQ

Microsoft Teams:
https://bit.ly/2ElmdXs

Facebook

Microsoft Teams

Slide aggiornate all'A.A. 2022-23

- **Ci si può presentare all'esame** solo se si è sostenuto ed è regolarmente presente su ESSE3:
  «Fondamenti di programmazione 1»,
  «Architettura degli elaboratori»

NO «Me lo verbalizza domani», «L'avevo quasi passato»,
«Devono uscire i risultati», «Mio nonno è stato colpito da un meteorite»

- Bisogna avere almeno seguito «Fondamenti di programmazione 2» (ex «Programmazione a oggetti»)
- Fortemente richiesta la conoscenza di un linguaggio a oggetti (meglio se Python)

- SISTEMI OPERATIVI:
  Prova scritta (con pre-test)

- SISTEMI OPERATIVI E RETI (corso disattivato): esame in due parti
  1. Scritto in laboratorio sul modulo di SISTEMI OPERATIVI (50% del voto. Possibile pre-test). La prova è identica a quella di SISTEMI OPERATIVI e si svolge nelle stesse date
  2. Orale con discussione progetto sul modulo di RETI DI CALCOLATORI (50% del voto)

* ORARIO
* DIFFICOLTÀ DELL'ESAME
* RICEVIMENTO, DIRITTI E DOVERI DELLO STUDENTE
* GESTIONE DELLE RACCOMANDAZIONI

Modalità d'esame e propedeuticità

Fondamenti di programmazione

Sistemi operativi

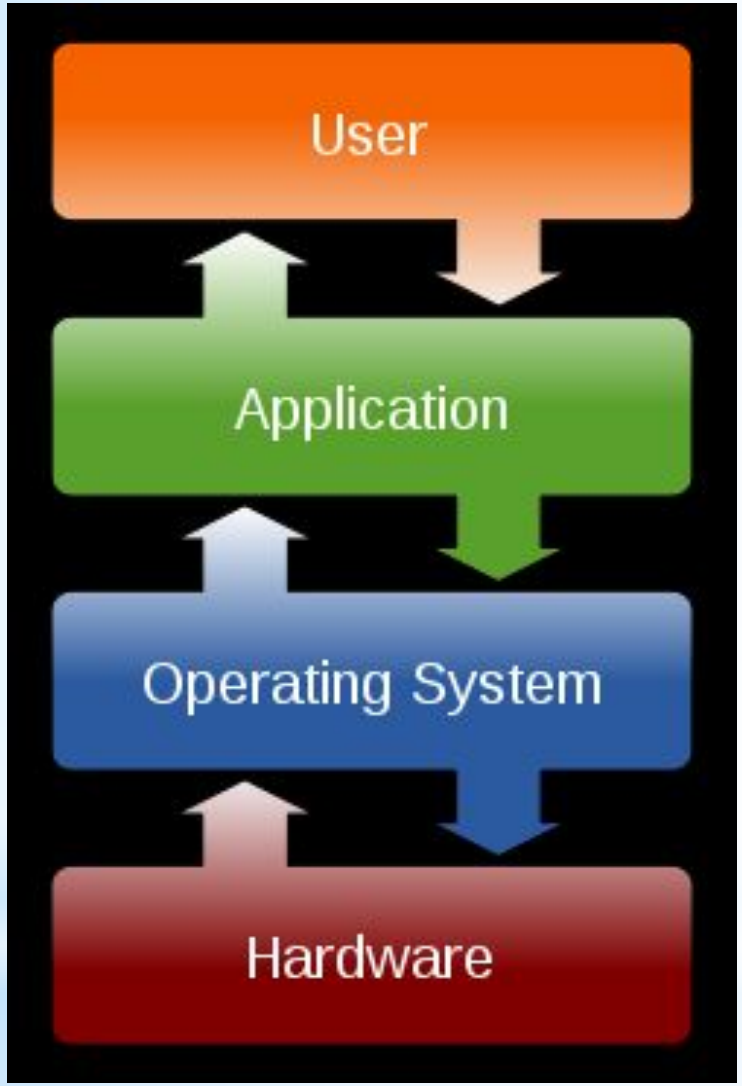
Propedeuticità

Difficoltà dell’esame

Programma del corso di Sistemi Operativi

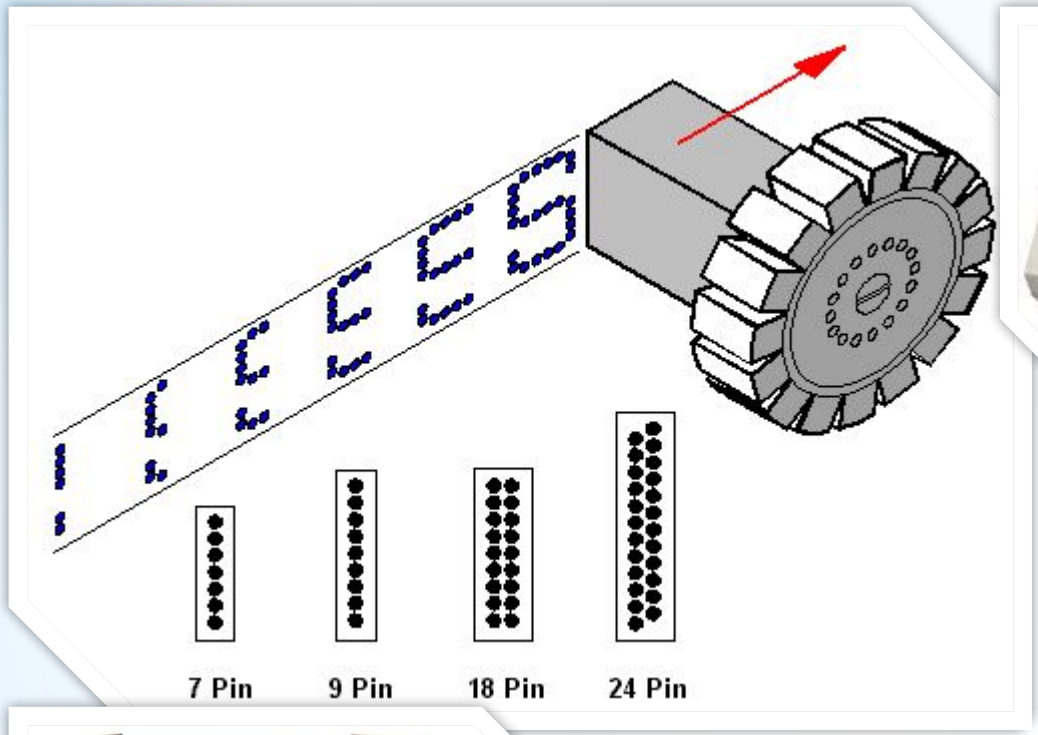
* **Programma del Modulo:**
  * Teoria dei sistemi operativi: gestione dei file, dei processi e della memoria.
  * Linguaggi di scripting: Perl
  * Programmazione multithreading (Python)
  * Case studies: Linux, Windows, Sistemi Operativi Mobili

* **Testi suggeriti e materiale:**
  * A. Silberschatz: Operating Systems Concepts (8 o 9 edizione). Addison-Wesley,
  * Deitel & Deitel, Operating Systems.
  * Python 3 Concurrency: https://docs.python.org/3/library/concurrency.html
  * Learning Perl, ed. O'Reilly.
  * Materiale disponibile sul sito del corso
  * The Java tutorial online: concurrency (https://docs.oracle.com/javase/tutorial/)
  * Concurrency: State Models & Java Programs, ed. Wiley



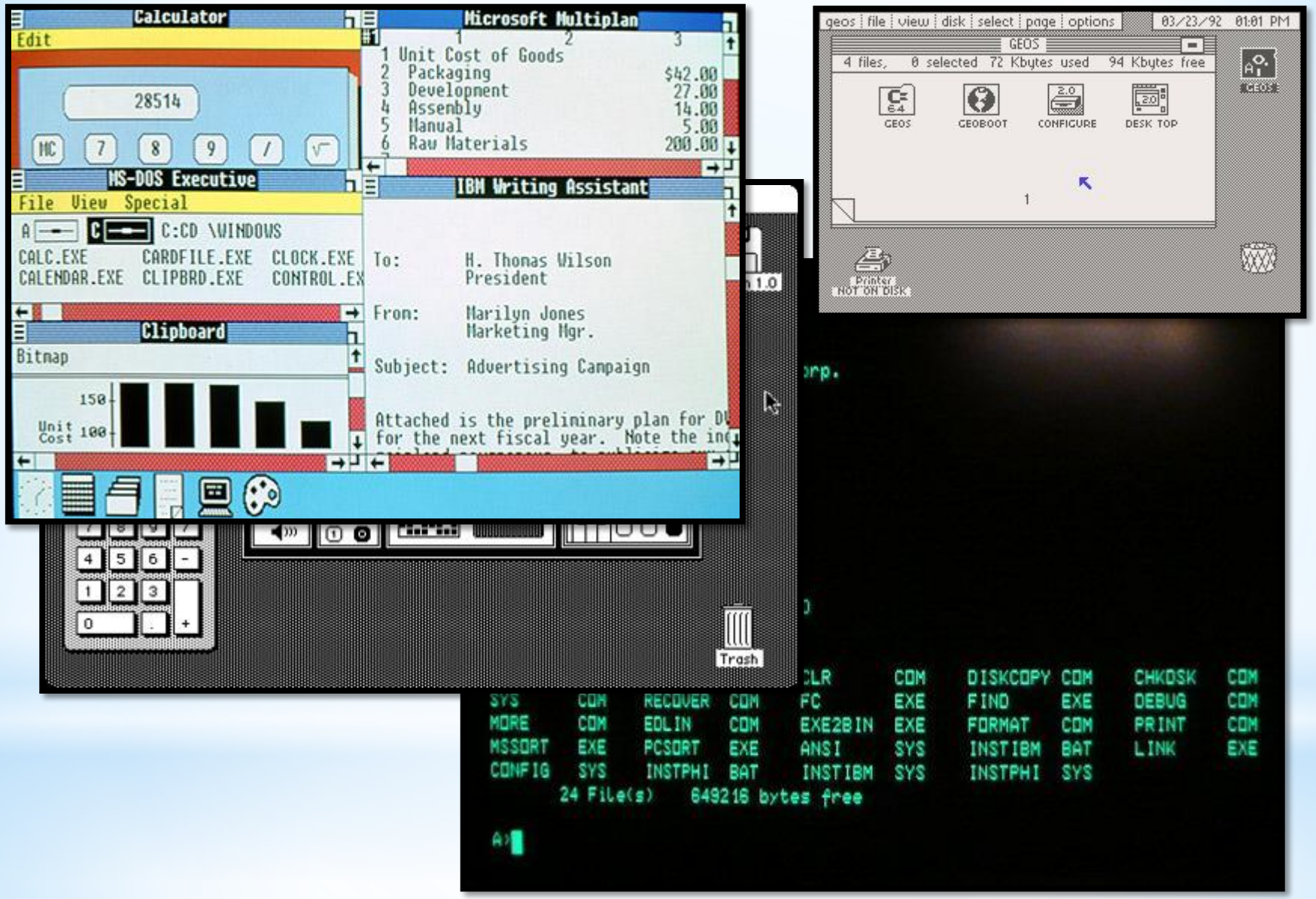
Visione astratta di un SO

*Tanto tempo fa sul pianeta Terra...

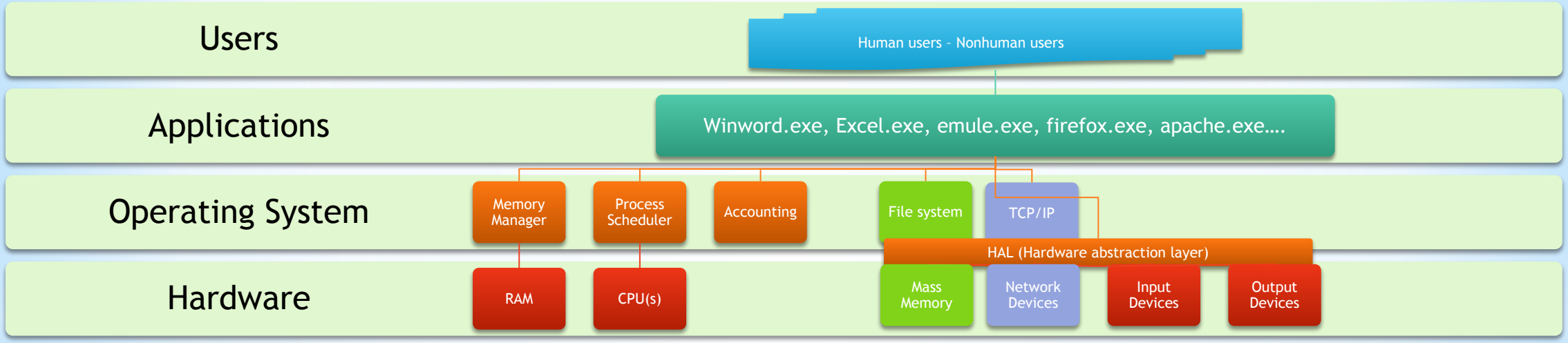


https://www.youtube.com/watch?v=Mopg8SMFuw0

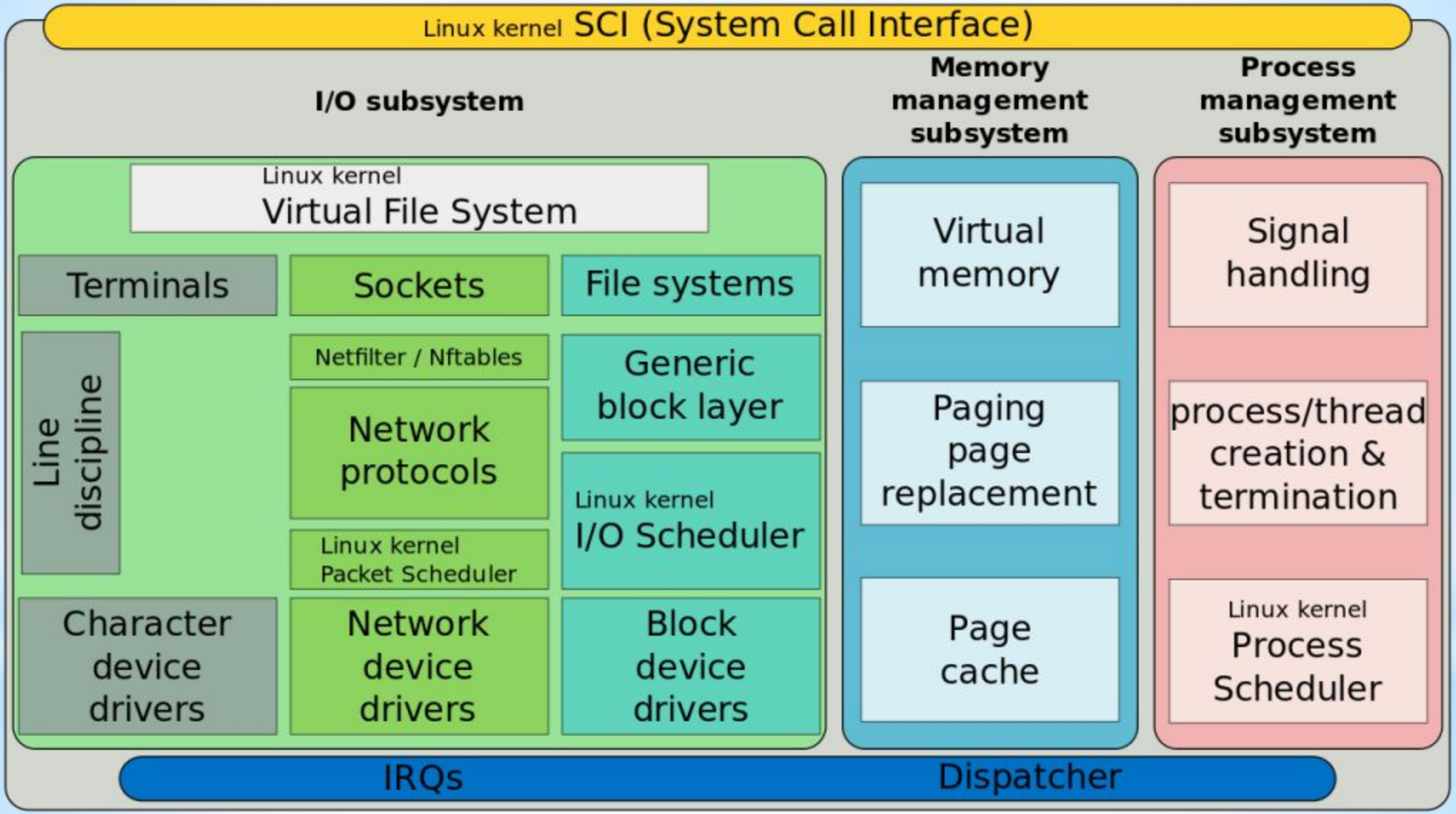
*Stampanti…



https://virtualconsoles.com/online-emulators/c64/



Visione di dettaglio


Linux kernel SCI (System Call Interface)
I/O subsystem
Memory management subsystem
Process management subsystem
Linux kernel
Virtual File System
Terminals
Sockets
File systems
Line discipline
Netfilter / Nftables
Network protocols
Linux kernel
Packet Scheduler
Generic block layer
Linux kernel
I/O Scheduler
Character device drivers
Network device drivers
Block device drivers
Virtual memory
Paging page replacement
Page cache
Signal handling
process/thread creation & termination
Linux kernel
Process Scheduler
IRQs
Dispatcher

Cos'è un sistema operativo

* Una ***raccolta*** di software predefiniti
* Scopo primario: *Gestire le **risorse** di un calcolatore*
  1. Generalizzare l'uso delle periferiche tramite un interfaccia comune
     - Funzioni di libreria comuni per le più grandi categorie di periferiche
  2. Gestire la memoria primaria (RAM)
     - Allocazione e Protezione
  3. Gestire la memoria secondaria (Dischi fissi, ecc.)
     - Mappatura sul dispositivo fisico
     - Organizzazione
     - Gestione degli errori
  4. **Gestire i processi e i thread**
     - **Scheduling, accesso condiviso alle risorse**
  5. Gestire gli eventi e l'interfaccia utente
     - Cattura e smistamento degli eventi ai processi
  6. Gestire i diritti di accesso e la presenza di più utenti
     - Accesso multiplo, quote, diritti di accesso
  7. Gestire la rete
     - Applicazioni distribuite, condivisione di risorse, socket

CreateFileA function (fileapi.h)

12/05/2018 • 29 minutes to read

Creates or opens a file or I/O device. The most commonly used I/O devices are as follows: file, file stream, di
volume, console buffer, tape drive, communications resource, mailslot, and pipe. The function returns a han
access the file or device for various types of I/O depending on the file or device and the flags and attributes

To perform this operation as a transacted operation, which results in a handle that can be used for transacte
CreateFileTransacted function.

Syntax

C++

```
HANDLE CreateFileA(
  LPCSTR                lpFileName,
  DWORD                 dwDesiredAccess,
  DWORD                 dwShareMode,
  LPSECURITY_ATTRIBUTES lpSecurityAttributes,
  DWORD                 dwCreationDisposition,
  DWORD                 dwFlagsAndAttributes,
  HANDLE                hTemplateFile
);
```

Parameters

`lpFileName`

The name of the file or device to be created or opened. You may use either forward slashes (/) or backslashe

In the ANSI version of this function, the name is limited to **MAX_PATH** characters. To extend this limit to 32,
the Unicode version of the function and prepend "\\?\" to the path. For more information, see Naming Files

For information on special device names, see Defining an MS-DOS Device Name.

fstat(2) - Linux man page

Name

stat, fstat, lstat - get file status

Synopsis

#include <sys/types.h>
#include <sys/stat.h>
#include <unistd.h>

int stat(const char *_path_, struct stat *_buf_);
int fstat(int _fd_, struct stat *_buf_);
int lstat(const char *_path_, struct stat *_buf_);

Feature Test Macro Requirements for glibc (see **feature_test_macros**(7)):

lstat():
_BSD_SOURCE || _XOPEN_SOURCE >= 500 || _XOPEN_SOURCE && _XOPE
|| /* Since glibc 2.10: */ _POSIX_C_SOURCE >= 200112L

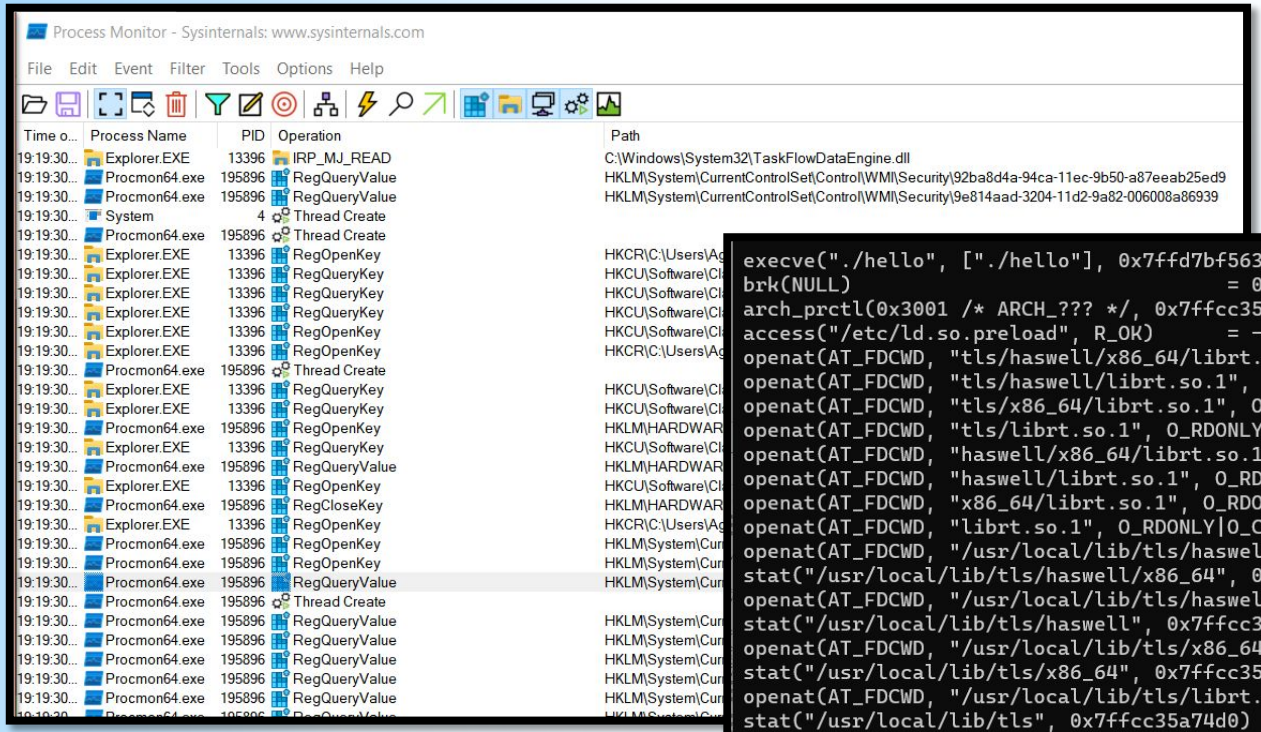
Description

These functions return information about a file. No permissions are required on th
of **stat**() and **lstat**() - execute (search) permission is required on all of the direct
the file.

stat() stats the file pointed to by *path* and fills in *buf*.

lstat() is identical to **stat**(), except that if *path* is a symbolic link, then the link it
that it refers to.

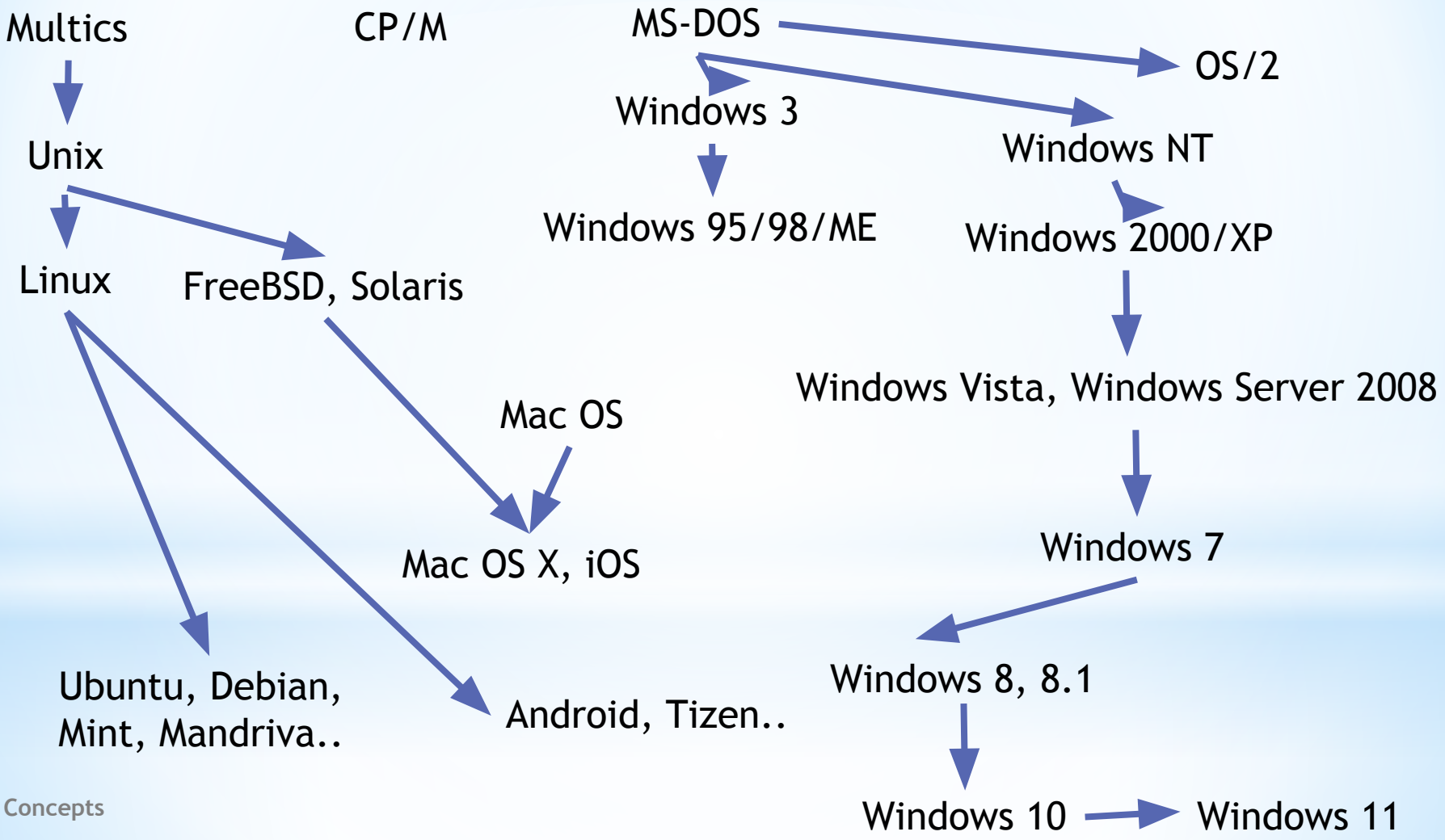
Funzioni di libreria

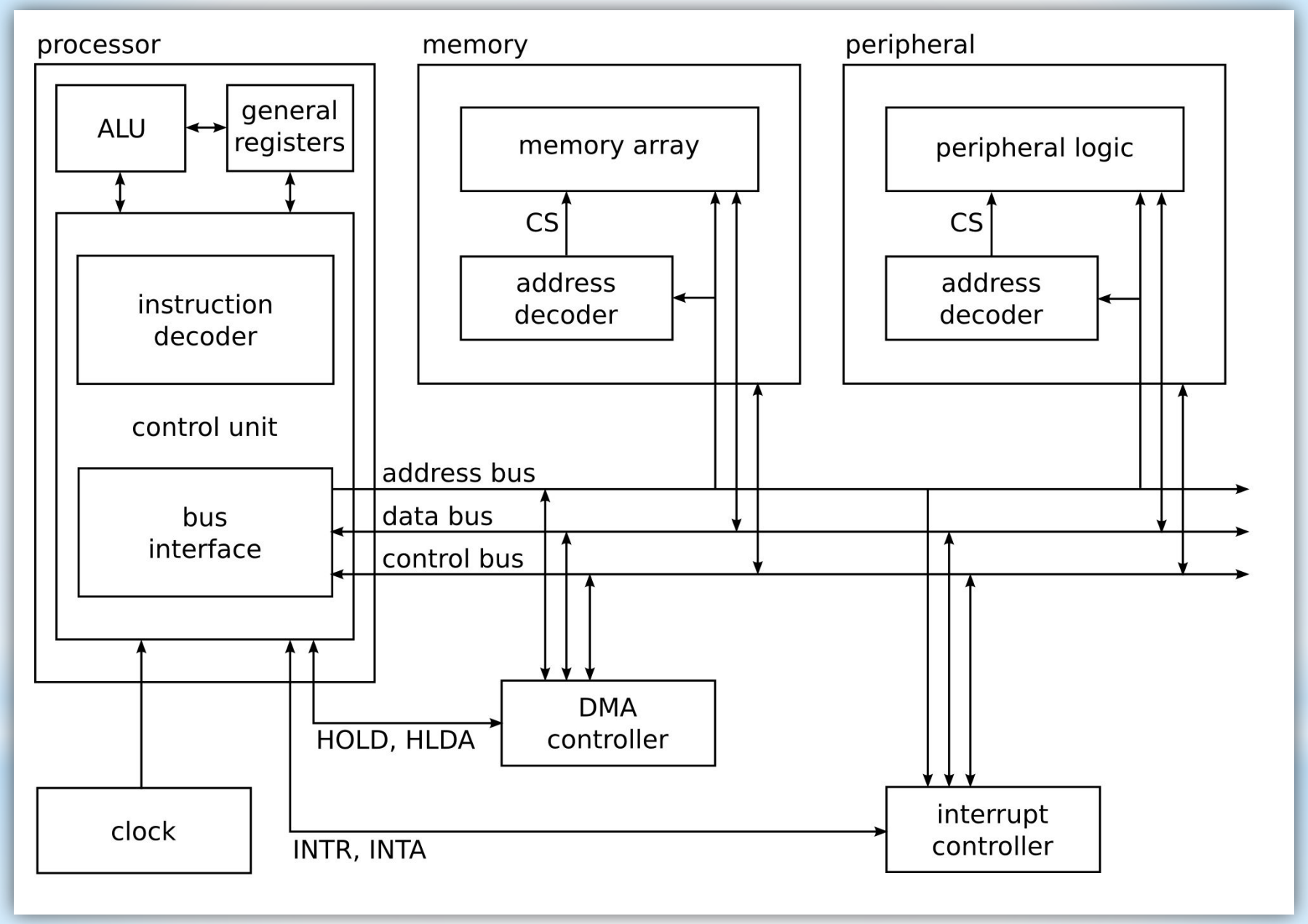



```
execve("./hello", ["./hello"], 0x7ffd7bf56360 /* 43 vars */) = 0
brk(NULL)                               = 0x559ccdc05000
arch_prctl(0x3001 /* ARCH_??? */, 0x7ffcc35a8280) = -1 EINVAL (Invalid argument)
access("/etc/ld.so.preload", R_OK)      = -1 ENOENT (No such file or directory)
openat(AT_FDCWD, "tls/haswell/x86_64/librt.so.1", O_RDONLY|O_CLOEXEC) = -1 ENOENT (No such file or directory)
openat(AT_FDCWD, "tls/haswell/librt.so.1", O_RDONLY|O_CLOEXEC) = -1 ENOENT (No such file or directory)
openat(AT_FDCWD, "tls/x86_64/librt.so.1", O_RDONLY|O_CLOEXEC) = -1 ENOENT (No such file or directory)
openat(AT_FDCWD, "tls/librt.so.1", O_RDONLY|O_CLOEXEC) = -1 ENOENT (No such file or directory)
openat(AT_FDCWD, "haswell/x86_64/librt.so.1", O_RDONLY|O_CLOEXEC) = -1 ENOENT (No such file or directory)
openat(AT_FDCWD, "haswell/librt.so.1", O_RDONLY|O_CLOEXEC) = -1 ENOENT (No such file or directory)
openat(AT_FDCWD, "x86_64/librt.so.1", O_RDONLY|O_CLOEXEC) = -1 ENOENT (No such file or directory)
openat(AT_FDCWD, "librt.so.1", O_RDONLY|O_CLOEXEC) = -1 ENOENT (No such file or directory)
openat(AT_FDCWD, "/usr/local/lib/tls/haswell/x86_64/librt.so.1", O_RDONLY|O_CLOEXEC) = -1 ENOENT (No such file or directory)
stat("/usr/local/lib/tls/haswell/x86_64", 0x7ffcc35a74d0) = -1 ENOENT (No such file or directory)
openat(AT_FDCWD, "/usr/local/lib/tls/haswell/librt.so.1", O_RDONLY|O_CLOEXEC) = -1 ENOENT (No such file or directory)
stat("/usr/local/lib/tls/haswell", 0x7ffcc35a74d0) = -1 ENOENT (No such file or directory)
openat(AT_FDCWD, "/usr/local/lib/tls/x86_64/librt.so.1", O_RDONLY|O_CLOEXEC) = -1 ENOENT (No such file or directory)
stat("/usr/local/lib/tls/x86_64", 0x7ffcc35a74d0) = -1 ENOENT (No such file or directory)
openat(AT_FDCWD, "/usr/local/lib/tls/librt.so.1", O_RDONLY|O_CLOEXEC) = -1 ENOENT (No such file or directory)
stat("/usr/local/lib/tls", 0x7ffcc35a74d0) = -1 ENOENT (No such file or directory)
openat(AT_FDCWD, "/usr/local/lib/haswell/x86_64/librt.so.1", O_RDONLY|O_CLOEXEC) = -1 ENOENT (No such file or directory)
stat("/usr/local/lib/haswell/x86_64", 0x7ffcc35a74d0) = -1 ENOENT (No such file or directory)
openat(AT_FDCWD, "/usr/local/lib/haswell/librt.so.1", O_RDONLY|O_CLOEXEC) = -1 ENOENT (No such file or directory)
stat("/usr/local/lib/haswell", 0x7ffcc35a74d0) = -1 ENOENT (No such file or directory)
openat(AT_FDCWD, "/usr/local/lib/x86_64/librt.so.1", O_RDONLY|O_CLOEXEC) = -1 ENOENT (No such file or directory)
stat("/usr/local/lib/x86_64", 0x7ffcc35a74d0) = -1 ENOENT (No such file or directory)
openat(AT_FDCWD, "/usr/local/lib/librt.so.1", O_RDONLY|O_CLOEXEC) = -1 ENOENT (No such file or directory)
stat("/usr/local/lib", {st_mode=S_IFDIR|0755, st_size=4096, ...}) = 0
openat(AT_FDCWD, "/home/ianni/.local/lib/tls/haswell/x86_64/librt.so.1", O_RDONLY|O_CLOEXEC) = -1 ENOENT (No such file or directory)
stat("/home/ianni/.local/lib/tls/haswell/x86_64", 0x7ffcc35a74d0) = -1 ENOENT (No such file or directory)
openat(AT_FDCWD, "/home/ianni/.local/lib/tls/haswell/librt.so.1", O_RDONLY|O_CLOEXEC) = -1 ENOENT (No such file or directory)
stat("/home/ianni/.local/lib/tls/haswell", 0x7ffcc35a74d0) = -1 ENOENT (No such file or directory)
openat(AT_FDCWD, "/home/ianni/.local/lib/tls/x86_64/librt.so.1", O_RDONLY|O_CLOEXEC) = -1 ENOENT (No such file or directory)
stat("/home/ianni/.local/lib/tls/x86_64", 0x7ffcc35a74d0) = -1 ENOENT (No such file or directory)
openat(AT_FDCWD, "/home/ianni/.local/lib/tls/librt.so.1", O_RDONLY|O_CLOEXEC) = -1 ENOENT (No such file or directory)
```

Meet Strace & Procmon

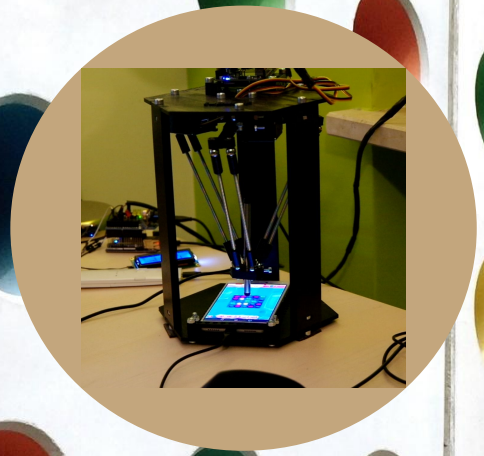
La storia


Multics
CP/M
MS-DOS
OS/2
Windows 3
Windows NT
Unix
Windows 95/98/ME
Windows 2000/XP
Linux
FreeBSD, Solaris
Windows Vista, Windows Server 2008
Mac OS
Windows 7
Mac OS X, iOS
Windows 8, 8.1
Ubuntu, Debian,
Mint, Mandriva..
Android, Tizen..
Windows 10
Windows 11


processor
memory
peripheral
ALU
general registers
memory array
peripheral logic
CS
CS
address decoder
address decoder
instruction decoder
control unit
bus interface
address bus
data bus
control bus
DMA controller
HOLD, HLDA
clock
INTR, INTA
interrupt controller

Elenco completo su
https://www.mat.unical.it/informatica/lanniGiovambattista

Temi di tesi e tirocinio



ROBOT

VIDEOGAME

SECURITY

http://bit.ly/3Y6baEt

https://www.youtube.com/watch?v=pgNjBhVs7_4