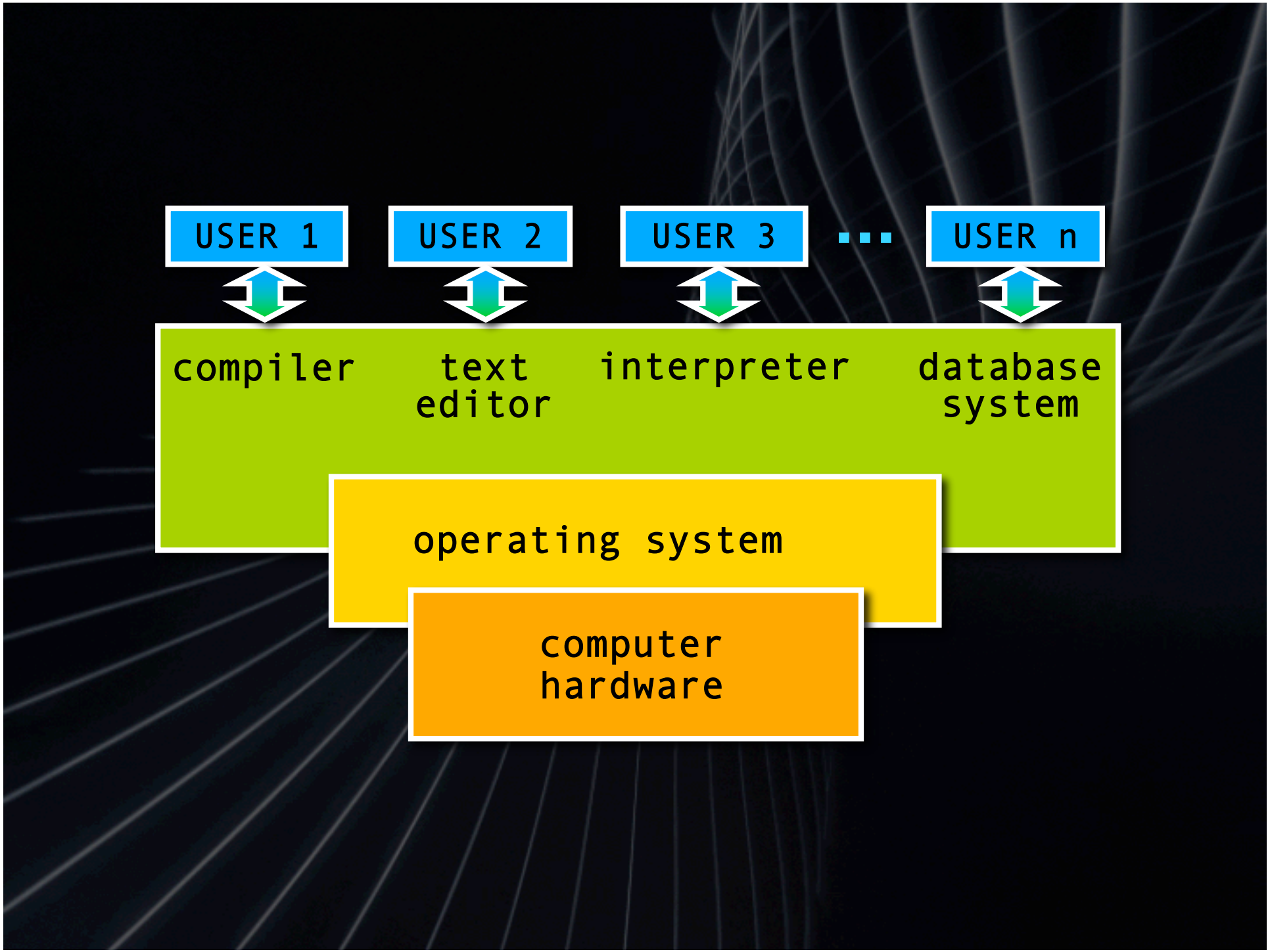

OPERATING
SYSTEMS OVERVIEW

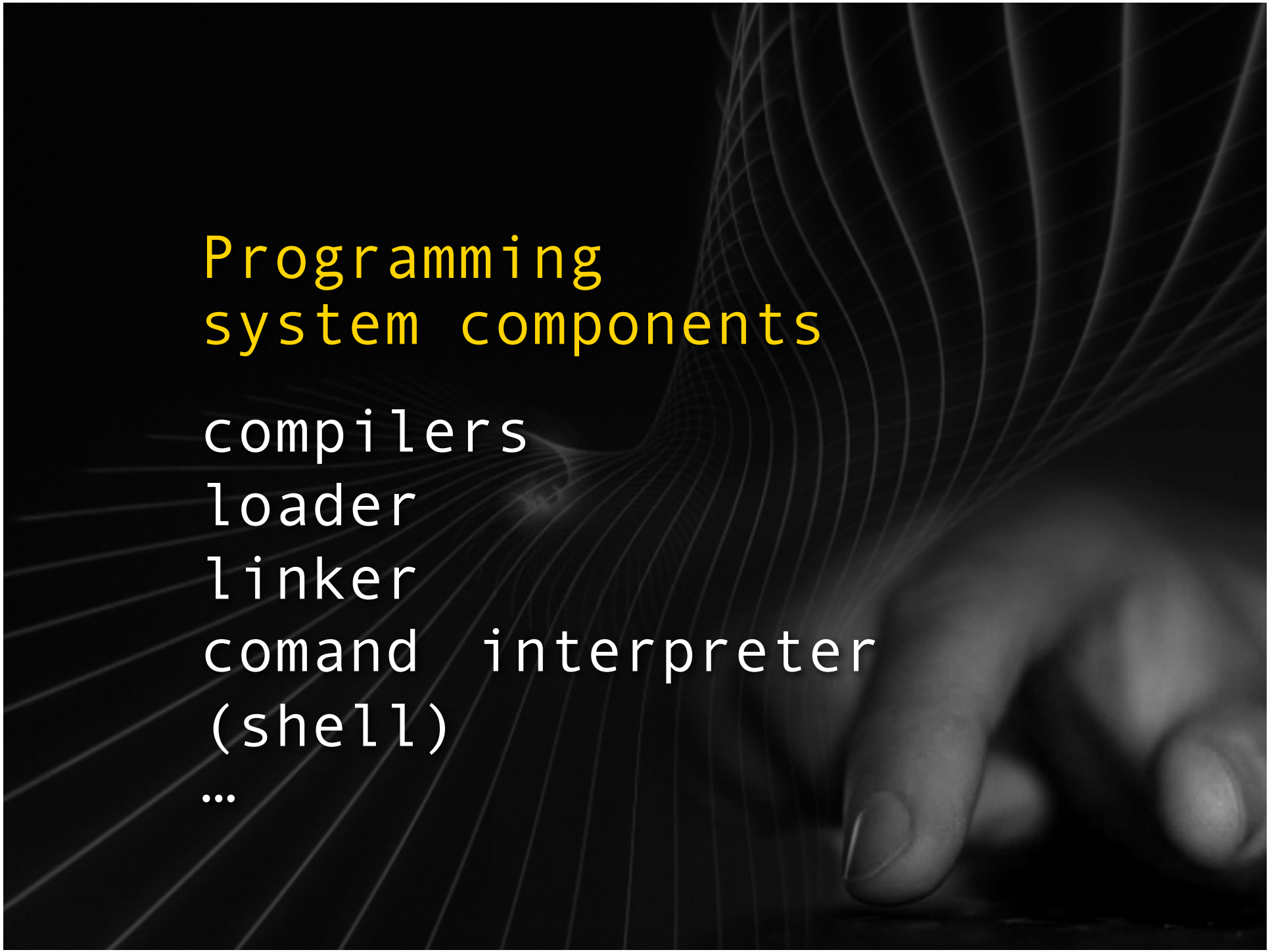
Contents

- O.S.Functions
- The Evolution of O.S.
- Characteristics of O.S.
- Basic hardware elements

Contents

- O.S.Components
- System calls
- O.S.Structure


USER 1
USER 2
USER 3
USER n
compiler
text editor
interpreter
database system
operating system
computer hardware


Programming
system components
compilers
loader
linker
comand interpreter
(shell)
…

O.S. purposes

- to make a computer more convenient and easier to use
- to allow more efficient operations of the whole computer system

To simplify the program development

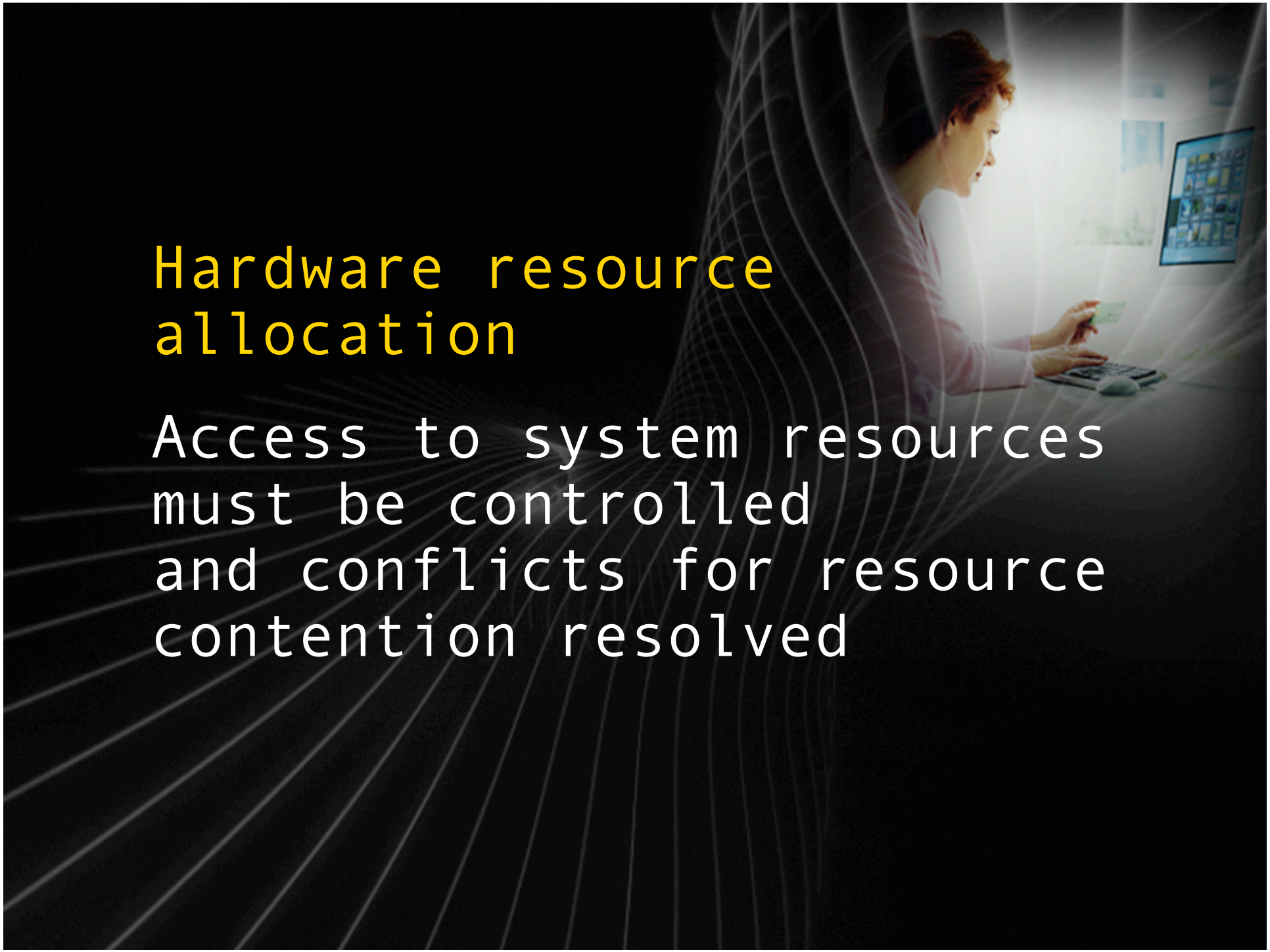
The O.S. masks the details of the hardware from the programmer and provides the programmer with a convenient interface for using system resources (system calls)

To simplify the program development

Definition of an extended (virtual) machine

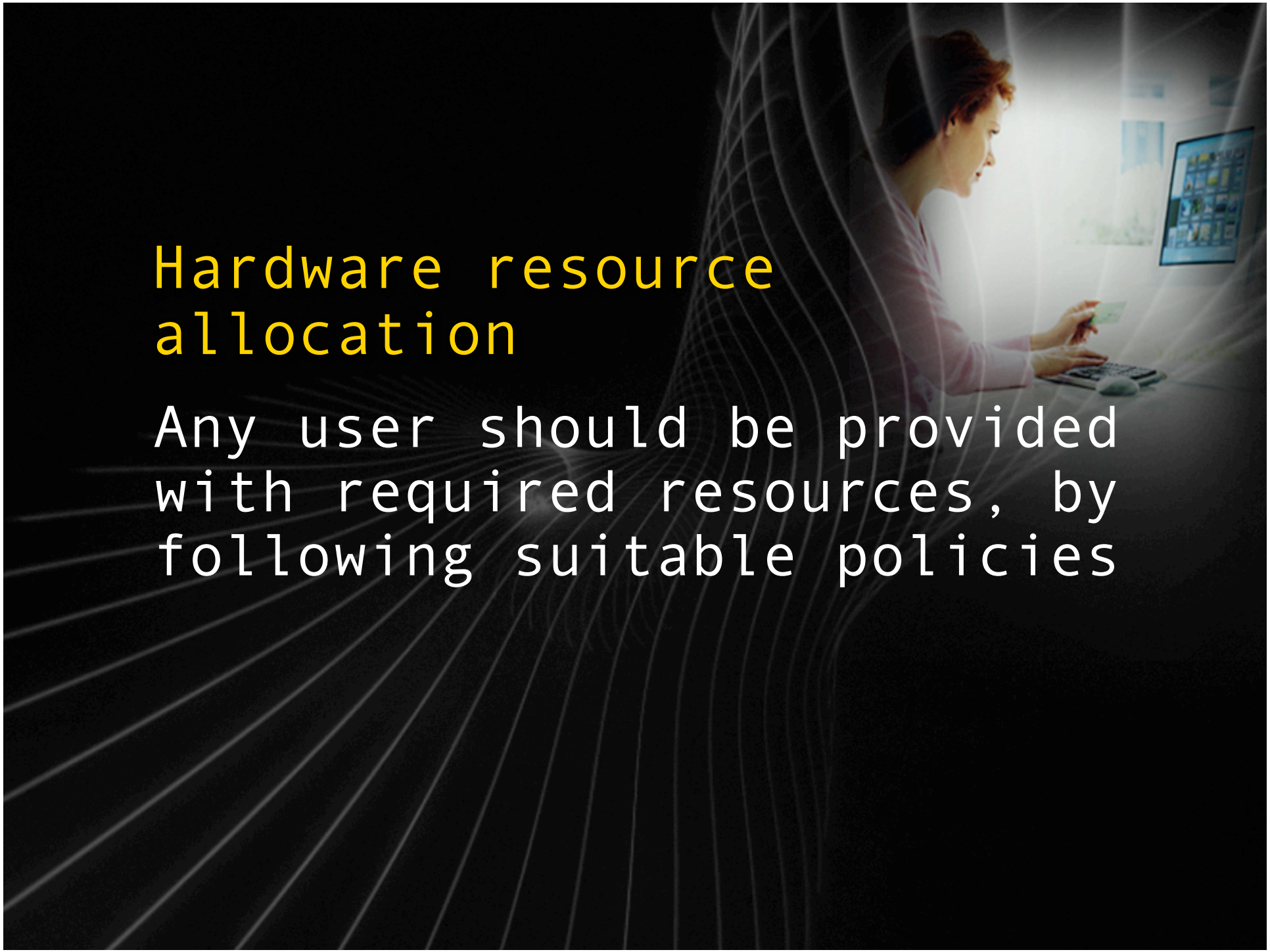
VIRTUAL MACHINE ES: DISK CONTROLLER

- commands: read, write, head motion, ecc…
- parameters: sector address, number of sectors for each track, ecc…
- state and error conditions



Hardware resource allocation

Access to system resources must be controlled and conflicts for resource contention resolved



Hardware resource allocation

Any user should be provided with required resources, by following suitable policies


The details for the management of hardware resources must be hidden to users

System calls provide the interface between the application programs and the O.S.


THE EVOLUTION
OF O.S.

Serial processing

- → No O.S.
- → Control by console
- → Scheduling
- → Setup time

Simple batch systems
Monitor

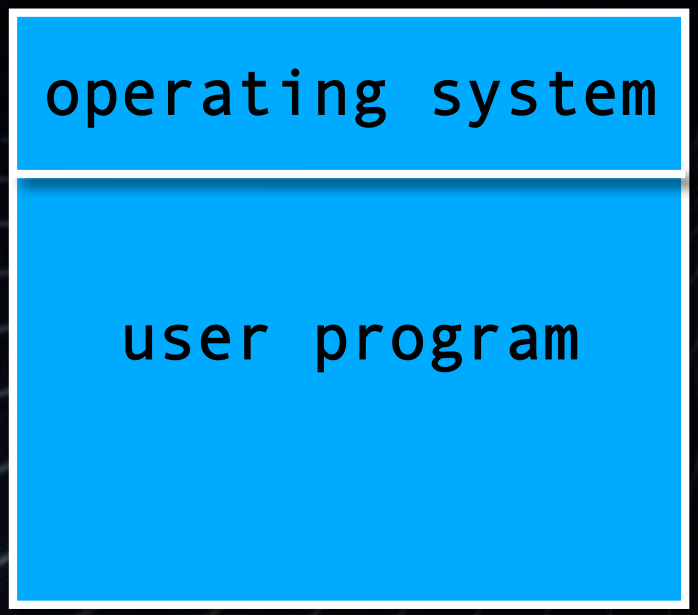
- → Resident in main memory
- → Control of the program execution
- → “batch” solution

Uniprogramming systems

Only one program
in main memory

Machine time alternates
between execution
of user programs
and monitor (OS)

Uniprogramming systems



main memory organization in a batch uni-programming system

Uniprogramming systems

Hardware and software resources of the computer system are dedicated to only one program (monouser system)

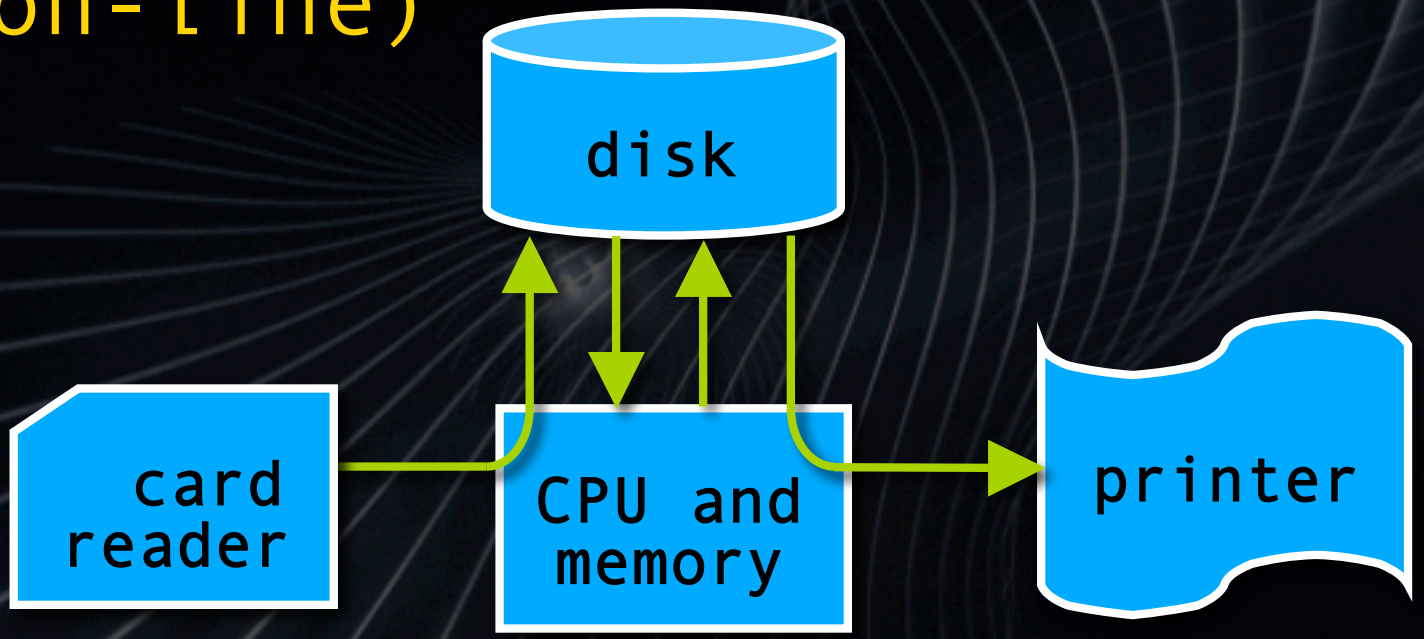
Uniprogramming systems

low CPU utilization

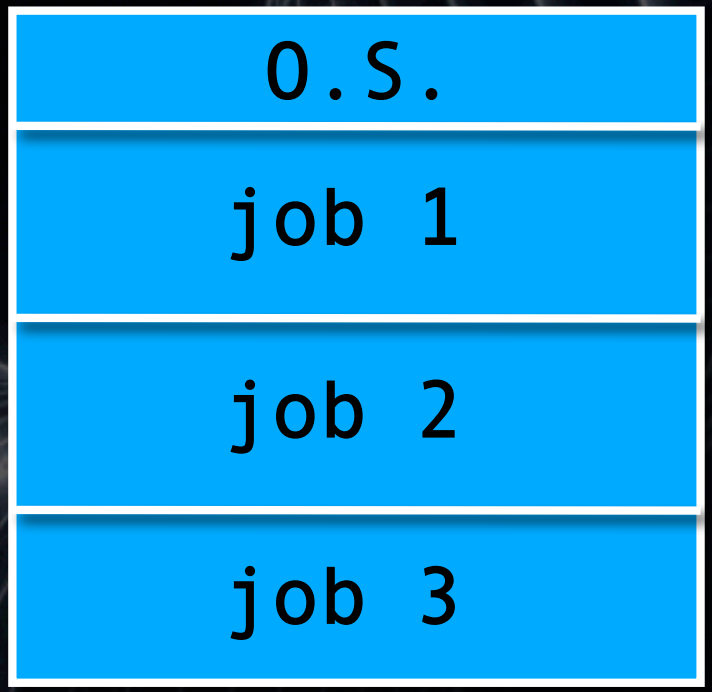
Hardware characteristics

- → Memory protection
- → Timer
- → Privileged instructions
- → Interrupts

Spooling (simultaneous peripheral operation on-line)



Multiprogrammed Batch Systems



A multiprogramming system with three jobs in memory

Multiprogrammed Batch Systems

Increase of resource utilization

Multiprogrammed Systems

O.S. are more sophisticated

→ Algorithms for resource management (CPU, memory, I/O)

Multiprogrammed Systems

O.S. are more sophisticated

→ Protection of the environments of different jobs

Multiprogrammed Systems

Scheduling algorithms

Job mix

- CPU-bound job
- I/O-bound job

Multiprogrammed Batch Systems

Example

OS/360 working on IBM 360 e 370 series

Multiprogrammed Batch Systems

- → High efficiency in resource utilization
- → Users cannot directly interact with the O.S.
- → Large response time

Interactive systems

→ direct communication between the user and the system

Interactive systems

→ the user gives instruction to the O.S. directly, by using either a keyboard or a mouse and waits for immediate results

Time-sharing systems

Logical extension of multiprogramming

→ Multiple users simultaneously access the system through terminals

Time-sharing systems

Logical extension of multiprogramming

→ The O.S. interleaves the execution of each user program in a short burst or quantum of computation

Time-sharing systems

At the end of the quantum (or during the quantum, if the job executes an I/O instruction) the CPU is switched to a different job.

Time-sharing systems

One of the first developed time sharing O.S. was the CTSS (Compatible Time Serie System), MIT years 60'

MULTICS, UNIX, …

Overhead