Introduction to the computer software

The operating systems

Software:
• set of instructions (program)
• set of data
• modification easy ("soft")

Multiple layers:

Hardware

Operating system

Application programs

Users

Tasks execution

Problem solving

Algorithms

Problem solving
Classification

From the hardware point of view: • software = set of instructions
  • either always in memory (resident)
  • either loaded on request (non-resident or transient)

From the user point of view: • Classification from the functionality

• System software:
  • Operating systems (including monitor, supervisor, …)
  • Loaders
  • Libraries and utility programs

• Support software (developers):
  • Assemblers
  • Compilers and interpreters
  • Editors
  • Debuggers

• Application software
The operating systems

Managing the resources and all the operations of the computer

• User interface
• Automatic calling of support and application programs

Basic tasks

(Performed by the operator on the 1st generation computers):

• Program instructions and data on the input device
• Memory reset
• Loading of the instructions and data into memory
• Program start (PC ← address of the 1st executable instruction)

Very slow manual operations with regard to the CPU cycle

→ progressive automatisation and

creation of operating systems
Control and program processing

Continuity in the succession of the tasks to be executed

example: simultaneity between task input and execution

The supervisor (monitor or kernel)

- Memory Resident
- The utility programs are stored on the secondary storage device
- Loaded into memory at power-on (bootstrapping)
- On request (user or automatically): tasks execution
- User interface: Job Control Language (JCL)

control instructions

- Instructions given as commands
- Graphics interface (icons, mouse,...)
Other control tasks

*Increase the system efficiency* → *suppress the dead times*

- **System scheduling tasks:**
  - **Multiprogrammation** *(concurrent programming)*
  - **Pseudo-parallel task treatment**
  - **Multiprocessing**
    - Multiple central units
    - Parallelism
    - **Dedicated processors** *(front-ends)*
- **Interrupts treatment**
- **Survey of the system status**
  - **Error detection** *(I/O devices, memory, …)*
  - **Priority definition**
  - **Security** *(file protection, intrusion control, …)*
The binary loader

- Part of the system resident software
- Loading the machine binary code of a task into memory
- Starting eventually the task
The utility programs

- sorting, merging, copying, inter-process communication, identification, ...

Support software

- The assemblers
  - Mnemonic identification of the basic instructions
  - Symbolic Addressing
- The high-level languages
  - Adapted to the type of application
  - Grammar (logic) easier than for the assemblers
  - Easier to read, to maintain (more natural language)
  - Interpreters ↔ compilers
- The debuggers
  - Identification of the execution errors (logic)
  - On-line (spy task)
  - Off-line (tracing task, memory dump, …)