

CHAPTER – 8

Abstraction for Programming:

- Most of the current commercial available multimedia applications are implemented in procedure-oriented programming languages (structural languages) such as C.
- In the part, multimedia specific functions (Example, changing the volume while playing an audio passage) were called, are respectively controlled, through hardware-specific libraries or device drivers.
- The application code of most commercial multimedia application programs are still highly dependent a hardware.
- Some applications are implemented with the help of tools.

Abstraction Levels:

Abstraction levels in programming define different approaches with a varying degree of details for representing, accessing and manipulating data.

- The abstraction levels with respect to multimedia data and their relations among each other are shown in below. A multimedia application may access each level

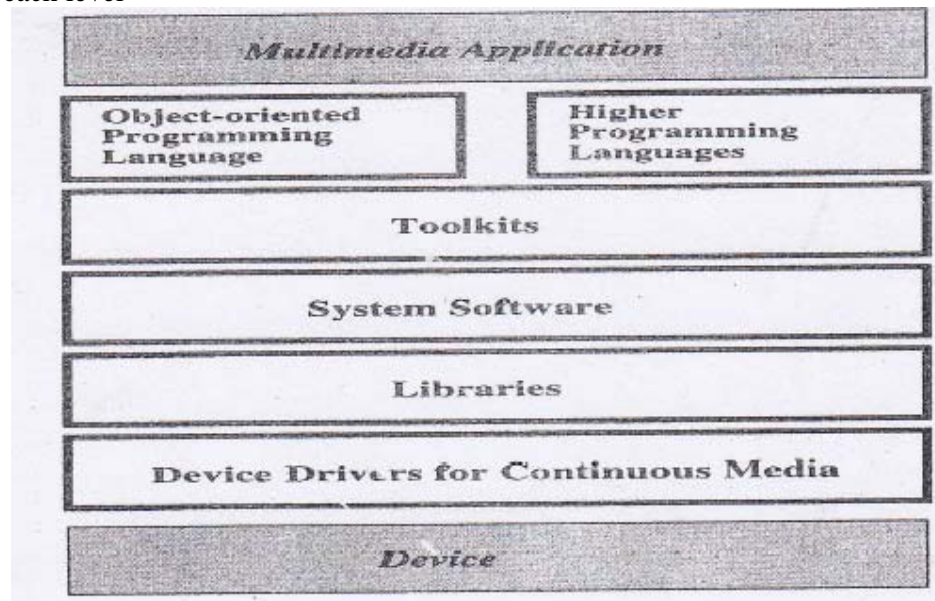


Figure: - Abstraction levels of the programming of multimedia system

Device:

- A device for processing continuous media can exist as a separate component in a computer. In this case, a device is not part of the operating system, but is directly accessible to every component and application.
- A library, the simplest abstraction level, includes the necessary functions for controlling the corresponding hardware with specific device access operations.

Device Drivers:

As with any device, multimedia device can be bound through a device driver, respectively with operating system. Hence, the processing of the continuous data becomes part of the system software.

Libraries:

The processing of continuous media is based on a set of functions which are embedded into libraries. This is the usual solution for programming multimedia data. These libraries are provided together with the corresponding hardware.

- The libraries differ very much in their degree of abstraction. Some libraries can be considered as extensions of the graphical user interface, where as other libraries consists of control instructions passed as blocks to the corresponding driver.
- Libraries are very helpful at the operating system level, but there is no agreement over which functions are best for different drivers, i.e. which functions should be supported.

System Software:

- Instead of implementing access to multimedia devices through individual libraries, the device access can become part of the operating system.
- An example of access to multimedia devices and support for continuous media processing. Implemented in operating system is the experimental Nemo system from the University of Cambridge. The Nemo system consists of the Nemo Trusted supervisor call, running in supervisor mode and three domains running in User mode: system, device driver and application.
- System software must appropriate schedulers such as rate scheduler or earliest-deadline: first scheduler. Multimedia device drivers embedded in operating systems simplify considerably the implementation of device access and scheduling.

Tool-Kits:

- A simplest approach (from the user perspective) in a programming environment than the system software interface for control of the audio and video data processing can be taken by using tool-kits.
- These tool-kits are used to –
 - o Abstraction from the actual physical layer (It is also done in a limited way by the libraries).
 - o Allow a uniform interface for communication with all different devices of continuous media (with eventual input of quality of service parameters).
 - o Introduce the client-server paradigm (here, the communication can be hidden from the application in an elegant way). (graceful, refined, tasteful).
- Tool-kits can also hide process structure.
- Tool-kits should represent interfaces at the system software level. In this case it is possible to embed them into the programming languages or object oriented environment.

Higher level Programming Language:

- Procedural higher level programming language will be called High Level Programming Language (HLL). In such an HLL, the processing of continuous media data is influenced by a group of similar constructed functions. These calls are mostly hardware and driver independent. Hence, their integration in HLLs leads to a wishful abstraction supports a better programming style and increases the productivity.
- Programs must be capable of supporting and effectively manipulating multimedia data. Therefore, the programs in HLL either direct access multimedia data structure or communicate directly with the active process in a real-time environment.

Object-oriented Analysis:

- The object-oriented approach was first introduced as a method for the reduction of complexity in the software development and it is used mainly with this goal today.
- Further, the reuse of software component is a main advantage of this approach.
- The basic ideas behind object-oriented programming are: data encapsulation and inheritance, in connection with class and object definitions.
- The programs are implemented, instead of using functions and data structures, by using classes, objects and methods.