



Chapter-10

Multimedia System (Pokhara University)



Scan to open on Studocu

Chapter 10 - Abstractions for Programming

This chapter focuses on different programming possibilities for accessing, and respectively representing multimedia data.

Abstraction Levels

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

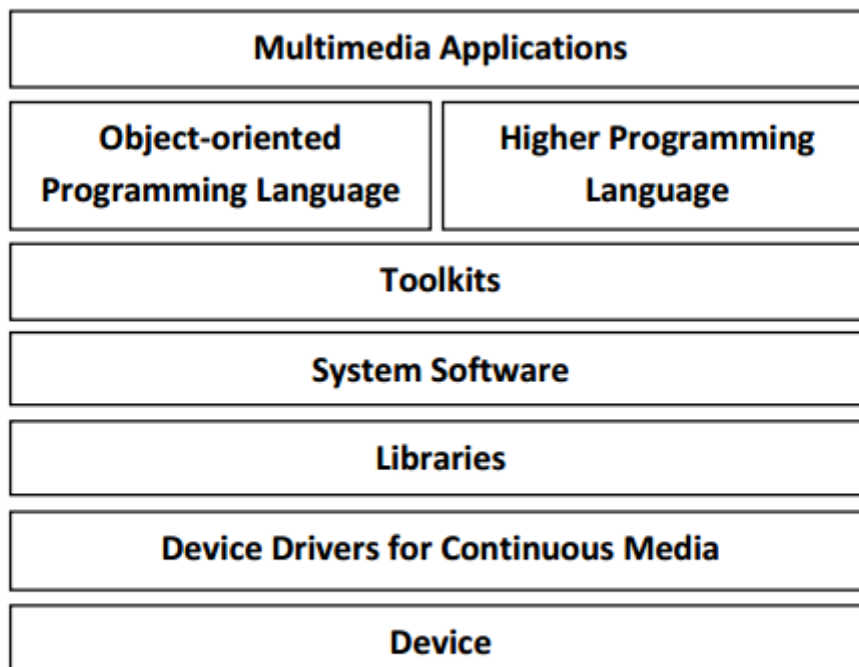


Figure: Abstraction levels of the programming of multimedia system

Device

A separate component in a computer that is used for processing continuous media.

Library

Simplest abstraction level, which includes the necessary functions for controlling the corresponding hardware with specific device access operations.

Device Drivers

It is used for bounding the multimedia devices.

System Software

It does the processing of continuous data. So, for this several properties are required like-schedulers (can be monotonic scheduler or earliest deadline-first scheduler).

Higher procedural programming language

Language used to implement multimedia applications contains abstractions of multimedia data.

Object-oriented programming language

Provides the application with a class hierarchy for the manipulation of multimedia.

Libraries

Libraries contain the set of functions used for processing the continuous media. Libraries are provided together with the corresponding hardware. Some libraries can be considered as extensions of the GUI, whereas other libraries consist of control instructions passed as control blocks to the corresponding drivers. Libraries are very useful at the operating system level. Since, there isn't any sufficient support of OS for continuous data and no integration into the programming environment exists, so there will always be a variety of interfaces and hence, a set of different libraries.

System Software

Instead of implementing access to multimedia devices through individual libraries, the device access can become parts of the OS. E.g. Nemo system.

The nemo system consists of the nemo trusted supervisor call running in the supervisor mode and 3 domains running in user mode: system, device drivers and application.

System processes implement the majority of the services provided by the OS. Devices processes are similar to system process, but are attached to device interrupt stubs which execute in supervisor mode. Application processes contains user programs.

Processes interact with each other via the system abstraction – IPC (InterProcess Communication). IPC is implemented using low-level system abstractions events and, if required, shared memory.

Data as time capsule:

Time capsules are the special abstraction related to the file systems. These files extensions serve as storage, modification and access for continuous media. Each logical data unit (LDU) carries in its time capsule, in addition to its data types and actual value, its valid life span. This concept is used widely in video than in audio.

Data as streams:

A stream denotes the continuous flow of audio and video data. A stream is established between source and sink before the flow. Operation on a stream can be performed such as play, fast forward, rewind and stop.

In Microsoft windows, a media control interface (MCI) provides the interface for processing multimedia data. It allows the access to continuous media streams and their corresponding devices.

Toolkits

Toolkits are used for controlling the audio and video data processing in a programming environment. Toolkit hides the process structures. It represents interfaces at the system software level. Toolkits are used to:

- ✓ Abstract from the actual physical layer.
- ✓ Allow a uniform interface for communication with all different devices of continuous media
- ✓ Introduce the client-server paradigm

Higher Programming language

In the higher programming languages, the processing of continuous media data is influenced by a group of similar constructed functions. These calls are mostly hardware and driver independent. The

programs in a high level language (HLL) either directly access multimedia data structures, or communicate directly with, the active processes in the real-time environment. The processing devices are controlled through corresponding device drivers.

Media can be considered differently inside a programming language.

- ✓ Media as types
- ✓ Media as files
- ✓ Media as processes
- ✓ Programming language requirements
 - Inter-process communication mechanism
 - Language

Media as Types

E.g. Programming expression used in OCCAM-2. This language is used for programming of transputers.

a, b REAL;

ldu.left1, ldu.left2, ldu.left-mixed AUDIO_LDU;

.....

WHILE

 COBEGIN

 PROCESS_1

 Input (micro1, ldu.left1)

 PROCESS_2

 Input(micro2,ldu.left2)

 ldu.left_mixed=a*ldu.left1+b*ldu.left2;

END WHILE

.....

One of the alternatives to programming is an HLL with libraries is the concept of media as types. In this example, there are 2 ldus from microphones that are read and mixed.

Media as files

Another possibility of programming continuous media data is the consideration of continuous media streams as files instead of data types.

File_h1=open(MICROPHONE_1,.....)

File_h2=open(MICROPHONE_2,.....)

File_h3=open(SPEAKER,.....)

.....

read(file_h1)

read(file_h2)

mix(file_h3, file_h1, file_h2)

activate(file_h1,file_h2, file_h3)

.....

deactivate(file_h1,file_h2, file_h3)

.....

rc1= close(file_h1)

rc2= close(file_h2)

rc3= close(file_h3)

This example describes the merging of two audio files.

Media as processes

The continuous media are mapped with the processes and then integrate them in an HLL.

```
PROCESS cont_Process_a;
```

```
.....
```

```
On_message_do
```

```
    Set_Volume.....
```

```
    Set_loudness.....
```

```
    .....
```

```
.....
```

```
[main]
```

```
pid=create(cont_process_a)
```

```
send(pid, set_volume,3)
```

```
send(pid, set_loudness)
```

```
.....
```

Here, the process cont_process_a implements a set of actions which apply to a continuous data stream.

Programming language requirements

HLL should support parallel processing. Number of processes must be known as compile time. Process should be defined dynamically at run-time.

Interprocess communication mechanism

- ✓ The IPC mechanism must be able to transmit the audio and video in a timely fashion because these media have a limited life span. The IPC must be able to:-
- ✓ Understand a prior and/or implicitly specified time requirements.
- ✓ Transmit the continuous data according to the requirements.
- ✓ Initiate the processing of the received continuous process on time.

Language

A simple language should be developed for the purpose of simplicity. An example of such language is OCCAM-2, ADA, parallel C-variant for transputer etc.

Object-Oriented Approaches

Object oriented approach is used to reduce the complexity in the software development. The basic idea behind object-oriented programming is data encapsulation & inheritance, in connection with class and object definition.

Abstract type definition

The definition of data types through abstract interfaces is called abstract type definitions. Abstract type definition is understood as an interface specification without a knowledge and implementation of internal algorithms. Data abstraction hides the used algorithms.

Class

The implementation of abstract data types is done through classes. A class specification includes an interface provided to the outside world.

Object

It is an instance of the class. All objects, derived from the same class include the same operations as an interface to the outside world. Object includes a set of operations, which are called methods. Object communicates among each other through the exchange of messages.

Inheritance

Classes contains root and leaves of the hierarchy, and also the super classes and subclasses (fathers and sons). Inheritance can be simple and multi inheritance.

Polymorphism

It is related to the property of inheritance indicating when the same name of a method is defined in several classes (& objects) with different implementation and functionalities. Polymorphism can cause programming errors that are difficult to find.

References:

- ✓ "Multimedia: Computing, Communications and Applications", Ralf Steinmetz and Klara Nahrstedt, Pearson Education Asia
- ✓ "Multimedia Communications, Applications, Networks, protocols ad Standards", Fred Halsall, Pearson Education Asia
- ✓ "Multimedia Systems", John F. Koegel Buford, Pearson Education Asia

Assignments:

- (1) Describe the Abstraction level. Explain briefly with the help of block diagram.
- (2) Describe Abstraction levels with respect to multimedia data and their relations among each other.
- (3) Show the different abstraction levels for programming of multimedia systems and explain the abstraction level of any one of them.