UNIT-V

v INTRODUCTION

• The operating system is the shield of the computer hardware against all software components.

• It provides a comfortable environment for the execution of programs, and it ensures effective utilization of the computer hardware.

• The operating system offers various serious related to the essential resources of a computer: CPU, main memory, storage and all input and output devices.

• For the processing of audio and video, multimedia application demands that humans perceive these media in a natural, error free way.

• These media originate from sources like microphone, etc and are transferred to destinations like loud speakers, etc and files located at the same computer or at remote station.

• Process Management

• Process management deals with the resource main processor. The capacity of this resource is specified as processor capacity.

• The process manager maps single processes onto resources according to a specified scheduling policy such that all processes meet their requirement.

• In most systems, a process u control of the process manager can adopt one of the following states:

• In the initial state, no process is assigned to program.

The process is in idle state.

• If a process is waiting for an event, i.e., the process lacks one of the necessary resources for processing, it is in the blocked state.

• If all necessary resources are assigned to the process, it is ready to run. The process only needs the processor for the execution of the program.

• A process is running as long as the system processor is assigned to it

• The process manager is scheduler. The component transfers a process into the ready state by assigning it a position in the respective queue of' the dispatches, which is that essential part of the operating system kernel.

• The dispatches manager the transitive from ready-torun to run. In most os, the next process to run is chosen according to a: priority policy. Between processes with the same priority, the one with the longest ready time is chosen.

v Real time Process Management in Conventional Operating Systems:

• An example: UNIX and its variants, MS-Windows-NT,Apple's systems 7 are the mostly and widely installed Operating systems with multitasking capabilities on personal Computer and workstation.

• Although enhanced with special Priority classes, it is not sufficient for multimedia application. For example, The SVR4 UNIX scheduler which provides a state priority is analyzed.

• For this, three Applications such as "typing" as interactive, "Video" as a Continuous media and a batch program. The result is additional features must be provided for the scheduling of Multimedia date processing. Let us see deeper into real time Abilities of one of these

v THREADS

•

• OS/2 was designed as a time- sharing operating system without taking serious real time application into account

• An OSt2 thread can be considered as alight-weight process: it is the unit of execution in the OS. A thread belongs exactly to one address spaces.

• All threads share the resources allocated and each has its own execution shack, register values and dispatch state. Such thread belongs to one of the following priority classes:

DEPARTMENT OF INFORMATION TECHNOLOGY IF-364

JAYAM COLLEGE OF ENGINEERI•NG AND TEHNOLOGY 4

• * The time critical class for threads that require immediate attention.

• * The fixed-high class is intended for good responsiveness applications.

• * The regular class is used for the executing if normal tasks.

• * The idle class consists of threads with lower priorities.

v Priorities

• Within each class, 32 different priorities (0,1 ,31) exist. Through time-slicing, threads of equal priority have equal chances to execute.

• The thread with a highest priority is dispatched and the time slicing is' started again.

• Time slice varies between 32 micro seconds and 65536 micro seconds.

• Threads are preemptive; the scheduler preempts the lower priority thread and assigns the CPU to the higher priority thread. The state of preempted thread is recorded so that execution can be resumed later.

• Physical device driver as process manager

• In OS/2, applications with real-tine requirement can run as physical device drivers at ring 0 (Kernel model).

• An interrupt that occurs on a desire can be serviced from the PDD immediately The PDD gets control to handle the interrupt as soon as it occurs.

• This may also include tasks running in ring 3 (user mode). The task running at ring 0 should leave the Kernel mode after 4 msec.

• PDD programming is complicated due to difficult tasking and debugging. it handles only request, regardless of

any other events.

• Different streams request real-time scheduling which are served by their PDDs.

• Internal time critical system activities cannot be controlled awl managed through PDDs. These PDDs are a solution for a system where streams arrive at only one device and no other activity is considered.

DEPARTMENT OF INFORMATION TECHNOLOGY IF-364

JAYAM COLLEGE OF ENGINEERI•NG AND TEHNOLOGY 5

v Enhanced system scheduler as Process Manager

• Time critical tasks can also be processed together with normal application running in ring 3, the user level.

• The critical tasks can be implemented by threads running in the priority class time-critical with one of the 32 priorities within this class.

• Each real-time task is assigned to one thread. A thread is interrupted if another thread with higher priority requires processing.

• Non-time critical applications run as threads are a regular class. They are dispatched by the OSScheduler according to their priorities.

• The main advantage of this approach is the control and co of all time critical threads through a higher instance.

v Meta Scheduler as Process Manager

• A Meta scheduler is employed to assign priorities to real-

4 time tasks

• Non-time critical tasks are processed when no timecritical task is ready for execution In an integrated system the process management of continuous data processes will not be realised as a meta scheduler. This method is applied in many UNIX systems.

v Real-time Processing Requirements.

• Continuous media data processing must occur in exactly predetermined-usually periodic-internals.

• Operations on this date reoccur over and over and must be completed at certain deadlines.

• The problem is to find a feasible scheduler which schedules all critical continuous media tasks in a way to meet deadlines.

• For scheduling of multimedia tasks, two conflicting goals must be considered:

• An uncritical process should not suffer from because time-critical processes are executed.

• Multimedia application really as much on text and graphs as an audio and video. Therefore, not all

resources should be occupied by the time critical process
and their management processes.
DEPARTMENT OF INFORMATION TECHNOLOGY IF-364
JAYAM COLLEGE OF ENGINEERI•NG AND TEHNOLOGY 6

• On the other hand, a time critical process must never be subject to priority inversion.

• The scheduler must ensure that any priority inversion is avoided or reduced as much as possible.

v Traditional Real-time scheduling

• The goal of traditional scheduling on time-saving computer is optional throughout optional resource utilization and fair queuing.

• In contrast, the main goal of real-time laster is to provide a schedule that allows all, respectively, as many time-critical processes as possible, to be processed in time, according to their deadline.

• The scheduling algorithm must map tasks on to resources such all tasks meet their time requirements.

• Therefore, it must be possible to show, or to prove, that a scheduling algorithm applied to real-time system fulfill the timing requirements of the task. There are several attempts to some real-time scheduling problems.

• To find best solution, two basic algorithms EDFA and

RMS algorithm are analyzed

v Real-time Scheduling: system model

• All scheduling algorithm to be introduced are based on the following system model for scheduling of real-time tasks. Their essential components are the resources tasks and scheduling goals.

• For multimedia systems, synchronized data can be processed by a. single process. The time constraints of a periodic task T are characterized by the following parameters.

• s Starting point

• e : Processing time of T

• d: Deadline ofT

• p : Period ofT

• r : Rate ofT(rVp)

• whereby

• (see figure)

DEPARTMENT OF INFORMATION TECHNOLOGY IF-364

JAYAM COLLEGE OF ENGINEERI•NG AND TEHNOLOGY 7

• The starting point S is the first time when the period task requires processing.

• Afterwards, it requires processing in every period with a processing time of e. At st(k-i)p, the task T is ready for kprocessing.

• The processing of T in period k must be finished at st(ki) p+d. For continuous media tasks, it is assumed that the deadline of the period (k-i) is the ready time of period k. This is known as congestion avoiding deadlines,

• The deadline for each message (d) coincides with the period of the respective periodic tasks (p).

• Tasks can be preemptive or non-preemptive. In real-time system, the scheduling algorithm must determine a schedule for an exclusive, limited resource that is used by different processes

• concurrently such that all of them can be processed without violating any deadlines. This notion can be extended to a model with multiple resources of the same type.

• A scheduling algorithm is said to guarantee a newly arrived task if the algorithm can find a schedule where the new task and all previously guaranteed tasks can finish processing to their deadlines in every period over the whole run-time.

• To guarantee tasks, it must be possible to check the schedulability of newly arrived tasks.

DEPARTMENT OF INFORMATION TECHNOLOGY IF-364

JAYAM COLLEGE OF ENGINEERI•NG AND TEHNOLOGY 8

• The major performance metric for a real-time scheduling algorithm is the guarantee ratio. It is the total number of guaranteed tasks versus the number of tasks which could be processed.

• Another performance metric is the processor utilization.

• This is the amount of processing time used by guaranteed tasks versus the total amount of processing time.

v Earliest Deadline First Algorithm

• The EDF algorithm is one of the best known algorithms for real-time processing. At every new ready state, the schedule selects the task with the earliest deadline among the tasks that are ready and not fully processed.

• the requested resource is assigned to the task, EDF must be computed immediately leading is a new order, i.e., the running task is preempted and the new task is scheduled according to its deadline.

• The new task is processed immediately if its deadline is earlier than that of the interrupted task.

• The processing of the interrupted task is continued according to the EDF algorithm later on.

• EDF is not only a algorithm for periodic tasks, but also for tasks with arbitrary requests, deadlines, and service execution 4 times. In this case, no guarantee about the processing of any task can be given.

• EDF is an optional, dynamic algorithm, i.e., it produces a valid schedule whenever one exists.

• A dynamic Algorithm schedules every instance of each incoming task according to specific demands.

• Tasks of periodic processes must be scheduled in each period again.

• With in tasks which have arbitrary ready times and deadlines, the complexity is o(n2)

• For a dynamic algorithm like EDF, the upper bound of the processor utilization is 100%. Compound with any static priority assignment, EDF is optional in the sense that if a set of tasks can be scheduled by an static priority assignment

DEPARTMENT OF INFORMATION TECHNOLOGY IF-364

JAYAM COLLEGE OF ENGINEERI•NG AND TEHNOLOGY 9

• It also can be scheduled by EDF. With a priority-driven system scheduler, each task is assigned a priority according to its deadline.

• The highest priority is assigned to the task with the earliest deadline; the lowest to the one with the furthest. With every accruing task, priorities might have to be adjusted

• Applying EDF to the scheduling of continuous media data on a single processor machine with priority scheduling, process priorities are likely to be arranged quite often.

Ix worst case, all processes should be rearranged.

v Rate Monitoring Algorithm

• The rate monotonic scheduling principle is an optional, static,

• Priority-driven algorithm for preempting periodic jobs .

• Optional in this context means that there is no other static algorithm that is able to schedule by the tasks set which cannot be scheduled by the rate monotonic algorithm.

• A process is scheduled by a static algorithm at the beginning of the processing. Subsequently, each task is processed with the priority calculated the beginning. No further scheduling is required.

• The following assumptions are necessary prerequisites to apply the rate monotonic algorithm.

• 1. The requests for all tasks with deadlines are periodic, i.e., have constant internals between consecutive requests.

• 2. The processing a single task must be finished before the next task of the same data stream becomes ready for

execution, Deadline consist of run ability constraints
only

• 3. All tasks are independent. This means that the requests for a certain task to not depend on the initiation or completion of requests for any other task.

• 4. Run-time for each request of a task is constant. Runtime denotes the maximum time which is requested by a processor to execute the task.

DEPARTMENT OF INFORMATION TECHNOLOGY IF-364

JAYAM COLLEGE OF ENGINEER• ING AND TEHNOLOGY 10

• 5. Any non-periodic task in the system has no required deadline. Typically, they initiate periodic tasks or are tasks for feature recovery. They usually displace periodic tasks.

• Not all these are mandatory to employ the algorithm, state properties are assigned to tasks according to their request rates.

• The priority corresponds to the importance of a task relative to other tasks. Tasks w higher requests will have higher priorities. The task with the shortest period gets the highest priority.

•

• The rate monotonic algorithm is simple method to

schedule time-critical, periodic tasks in the respective
resource. The task always meets the deadline, it can be
proven for longest response time, and the response time
is the time span between the request and the end of the
processing task.

• This time span is maximal when all process with highest
priority at a same time.

• This case is critical instant. In the figure, the priority of a
is, according to RM algorithm, higher than b, and b is
higher than c. The critical time zone is the time internal
between critical instant and the completion of a task

DEPARTMENT OF INFORMATION TECHNOLOGY IF-364

JAYAM COLLEGE OF ENGINEER• ING AND TEHNOLOGY 11

v Pre-emptive Versus Non-preemptive Task
Scheduling

• Real time tasks can be distinguished into preemptive
and non-preemptive tasks. If a task is non-preemptive, it
is finished or requires further resources.

• The process of preemptive task is interrupted by
highest-priority task. In most cases where tasks are
treated as non-preemptive parameters are unknown
until task arrives.

• The best algorithm is the one which maximizes the number of completed tasks.

• To guarantee the processing of periodic process and to get a feasible schedule for a periodic task set, tasks are usually heated as preemptive.

• One reason is that the high preemptabiity minimizes priority inversion. Another reason is that for some nonpreemptive tasks sets, no feasible schedule can be found; whereas for preemptive scheduling, it is possible. Figure shows an example where the scheduling of preemptive tasks is possible, but non-preemptive tasks cannot be scheduled.

• A task set of m periodic, preemptive tasks with processing times ei and request periods with Fixed priority assignment

• Here, the preemptive of tasks is a necessary prerequisite to check their schedulability.

DEPARTMENT OF INFORMATION TECHNOLOGY IF-364

JAYAM COLLEGE OF ENGINEER• ING AND TEHNOLOGY 12

v FILE SYSTEMS

• File system is the most visible part of an operating system Most programs write or read files Their program code, as well as user data, are stored in files.

• The organization of the file system is an important factor for the usability and convenience of operating system.

• A file is a sequence of information held as a unit for storage and use in a computer $ system.

• Files are stored in secondary storage. In traditional file systems, information types stored in files are sources, objects, libraries and executables of programs, numeric data, text, payroll records, etc. In multimedia systems, the stored information also covers digitized video and audio with their related real-time "read" and "write" demands.

v Traditional File Systems

• The two main goals are

• (1) To provide a comfortable interface for file access to the user and

• (2) To make efficient usage of storage media.

• Whereas the first goal is still an area of interest for research the structure, organization and access of data stored on disk have been extensively discussed and investigated over the last decades.

• To understand the specific multimedia developments in this area, this section gives a brief overview on file, file system organizations and file access mechanism.

v File structure

• We commonly distinguish between two methods of file organization. In sequential storage each file is organized as a simple sequence of bytes or records. Files are stored on sequent on the secondary storage media as shown in figure.

DEPARTMENT OF INFORMATION TECHNOLOGY IF-364

JAYAM COLLEGE OF ENGINEER• ING AND TEHNOLOGY 13

• They are separated from each other by a well defined "end of file" bit pattern, character or character sequence.

• A file description is usually placed at the beginning of the file and is, in some systems, repeated at the end of the file. Sequential storage is the only possible way to organize the storage on tape, but it can also be used on disks. The main advantages are its efficiency for sequential access, as well as for direct access.

• Disk access time for reading and writing is minimized.

• Additionally for further improvement of performance with caching, file can be read a head of the user program.

• In systems where file creation, deletion and size modifications occur frequently, sequential storage has major v Disadvantages.

• In non-sequential storage, the data items are stored in a son-contiguous order. There exist mainly to approaches.

• *One way is to use linked blocks where physical blocks containing consecutive logical locations are linked using pointer.

• The file descriptor must combine the number of blocks occupied by the file, the pointer to the first block and it may also have the pointer to the last block is the cost of the implementation for random access because all prior data must be read.

DEPARTMENT OF INFORMATION TECHNOLOGY IF-364

JAYAM COLLEGE OF ENGINEER• ING AND TEHNOLOGY 14

• In MS-Dos, a similar method is applied. A File Allocation Table (FLA) is association with each disk. One entry in the table represents one disk block.. The directory entry of each file block. The number in the slot of an entry refers to the next block of a file. The slot of the last block of a file contains an end-of-file mark.

• *Another approach is to store block information in mapping tables. Each file is associated with a table where, apart from the block number, information like owner, file size, creation time, last access time, etc., are stored.

• Those tables usually have a fixed size, which means that the number of block references is bounded. Files with, more blocks are referenced indirectly by additional tables assigned to the files.

• In UNIX, a small table called and -node is associated with each file (see fig.).

• The indexed sequential approach is an example for multilevel mapping; here, logical and physical organization are not clearly separated.

DEPARTMENT OF INFORMATION TECHNOLOGY IF-364

JAYAM COLLEGE OF ENGINEER• ING AND TEHNOLOGY 15

v Directory Structure:

•

• Files are usually organizes in directions. Most of today's operating systems provide tree-structural directions where the user can organize the files according to his/her personal needs. In multimedia systems, it is important to organize the files in a way that allows easy, fast, and contiguous data access.

•

v Disk Management

• Disk access is a slow and costly transaction. In traditional systems, a common technique to reduce disk access is block caches. Using a block cache, blocks are kept in memory because it is expected that future read or write operations access these data again.

• Thus, performance is to reduce disk arm motion. Blocks that are likely to be accessed in sequence are placed together on the cylinder.

• To refine this method, rotating positioning can be taken into account. Consecutive blocks are placed on the same cylinder, but in an interleaved ways as shown in fig.,

• Another important issue in the placement of the mapping ables on the disk. If they are placed near the beginning of the disk, the distance between them and the blocks with be, on average, half the number of

DEPARTMENT OF INFORMATION TECHNOLOGY IF-364

JAYAM COLLEGE OF ENGINEER• ING AND TEHNOLOGY 16

cylinders. To improve this, they can be placed in the middle of the disk. Hence, the average seek time is roughly reduced by a factor of two. In the sane way, consecutive blocks should be placed on the same cylinder.

• The use of the same cylinder for the storage of mapping tables and referred blocks also improves performances.

v Disk Scheduling:

• Whereas strictly sequential storage devices (eg., tapes) do not have a scheduling problem, for random access storage devices, every file operation may require movements of the read/write head.

• This operation, known as “to seek”, is every time consulting, i.e, a seek time in the order of 250ms for CDs is still state of the art. The actual time to read or write a disk block is determined by:

* The seek time

* The latency or rotational delay.

* The actual data transfer time needed for the data to copy from disk into main memory.

• Usually the seek time is the largest factor of the actual transfer time. Most systems try to keep the cost of seeking) low try applying special algorithms to the scheduling of disk read/write operation.

• The access of the storage device in a problem greatly influenced by the fill allocation method.

• For instance, a program reading a contiguously allocated file generates requests which are located close together on a disk.

• Thus head movement is limited. Linked or indexed files

with blocks, which are widely scattered, cause many
head movements.

• In Multi-programming systems, where the disk queue may often be non-empty, fairness is also a criterion for scheduling! Most systems apply one of the following scheduling algorithms:.

DEPARTMENT OF INFORMATION TECHNOLOGY IF-364

JAYAM COLLEGE OF ENGINEER• ING AND TEHNOLOGY 17

• Shortest-Seek-Time First(SSTF):

• At every point in time, when a data transfer is requested, SSTF detects among all requests the one with the minimum seek time from the current head position.

• Therefore, the head is moved to the closest track in the request queue. This algorithm was developed to minimize seek time and it is in this sense optimal.

• SSTF is a modification of shortest job first (SJF), and like SJF, it may cause starvation of some requests.

• requests in the inner most and outer most dips areas.

Fig., demonstrates the operation of the STTF algorithm.

DEPARTMENT OF INFORMATION TECHNOLOGY IF-364

JAYAM COLLEGE OF ENGINEER• ING AND TEHNOLOGY 18

•

v Multi-media File Systems:

• Compared to the increased performance of processors and networks, storage devices have become only marginally faster.

• The effect of this increasing speed mismatch is the search for new storage structure, and storage and retrieval mechanisms with respect to the file system. Continuous media data are different from Desiree data in:

v Read Time Characteristics:

• As mentioned previously, the retrieval, computation and presentation of continuous media is time-dependent. The data must be presented before a well-defined deadline with small jitter only.

• Thus, algorithm for the storage and retrieval of such data must consider time constraints, and additional buffers to smooth the data stream must be provided.

v File Size:

• Compared to text and graphics video and audio have very large storage space requirements.

• Since the file system has to store information ranging from small, unstructured units like text flies to large, highly structured data units like video and associated

audio, it must organize the data on disk in a way that

DEPARTMENT OF INFORMATION TECHNOLOGY IF-364

JAYAM COLLEGE OF ENGINEER• ING AND TEHNOLOGY 19

efficiently uses the limited storage. For example, the

storage requirements of compressed CD quality stereo

audio one 1.4 M bits/s; low but acceptable quality

compressed video still requires about 1 M bits/s using,

eg., MPEG-i.

v Multiple Data Streams

• A multimedia system must support different media at

one time. It does not have to ensure that all of them

get a sufficient share of the resources, it also must

consider tight relations between different streams

arriving from different sources.

• The retrieval of a movie, for example, requires the

processing and synchronization of audio and video.

In the next section, a brief introduction to storage

devices are given. Then the organization of files on disks is

discussed.

v storage Devices:

• The storage subsystem is a major component of any

information system. Due to the immingle storage space requirements of continues media, convention magnetic storage devices are often not sufficient.

• Tapes, still in use in some traditional systems are inadequate for multimedia systems because they cannot provide independent accessible streams, and random access is slow & expensive.

v Multimedia Database Management System:

• Multimedia applications often address file management interfaces at different levels of abstraction. Consider the following three applications:

(1) a hypertext

(2) an audio editor

(3) an audio-video distribution service.

• All three doesn't have much in common, but all three can uniformly perform with MDBMS.

• MDBMS is associated between the application demark and device domain. The MDBMS is integrated into systems domain through the operating system and communication components.

• Therefore, all three applications can be put on the same abstraction level with respect to DBMS. Further, DBMS provides other properties in addition to storage abstraction:

DEPARTMENT OF INFORMATION TECHNOLOGY IF-364

JAYAM COLLEGE OF ENGINEER• ING AND TEHNOLOGY 20

• Persistence of data

• Consistent view of data

• security of data

• Query and retrieval of data

• An MDBM5 can be characterized by its objection when

handling multimedia data:

o Corresponding storage media

§ Description search methods

§ Device independent interface

§ Format independent interface

§ View specific and simulation data

§ Management of large amount of data

§ Relational consistency of data management

§ Real time data transfer

v The functions of the systems components around

MDBS:

• The operating system provides the management

interface for MDBS to all local devices

• The MDBMS provides an abstraction of the stored

• Data and their equivalent devices.

• The communication system provides for MDBMs

• Abstraction for communication with entities at remote computers.

• A layer above the DBMS, OS and communication system can unify all these different abstractions and offer them, for e.g.., in an object oriented environment such as tool kits. Thus an application should have access to each abstraction at different levels.

v DATA STRUCTURE FOR STORAGE

• In general, data can be stored in database either in unformatted form or in formatted form. Unformatted or unstructured data are presented in a unit where the content cannot be retrieved by accessing any structural detail.

• Formatted or structured data are stored in variables, fields or attributes with 0 values. Here, the particular data parts are characterized according to their properties.

DEPARTMENT OF INFORMATION TECHNOLOGY IF-364

JAYAM COLLEGE OF ENGINEER• ING AND TEHNOLOGY 21

v Raw Data

• An uncompressed image consists of a set of individual pixels. The pixels represent raw data in the form of bits and bytes. They create the unformatted information units, which represent a long sequence or set of symbols pixels, sample values, etc

• Registering Data

• To retrieve and correctly interpret such an image, the details of the coding and the size of the image must be known.

• Let as assume that each pixel in an image is encoded with eight bits for the luminance and both chrominance difference signals. The resolution will be 1O24x1O pixels.

• These registering data are necessary to provide a correct interpretation of the raw data. Traditional DBMSS usually know only numbers and characters, which have fixed semantics; therefore, no additional description is required. Image, audio and video data allow for a number of attributes during coding and 5 Without this additional description, the multimedia data could only be interpreted

v Descriptive Data

• Today, the search for textual and numerical content is very effective. However, the search for image, audio or video information is much more difficult. Therefore, optimal description should be described in the form of

text. These descriptive data provide additional
redundant information and ease data retrieval during
later searches. Descriptive data could be presented in
unstructured or structured form.

v Operations on Data (Information Retrieval)

• An MDBMS must offer corresponding operations for
archival and retrieval. The media-related operations will
be handled as part of or an extension of query
languages. In databases, following different classes of
operations of each medium are needed: input, output,
modification, deletion, comparison and evaluation.

DEPARTMENT OF INFORMATION TECHNOLOGY IF-364

JAYAM COLLEGE OF ENGINEER• ING AND TEHNOLOGY 22

• The input operation means that data will be written to
the database. In this case, the ra* and registering data
are always are needed.

• The descriptive data can be attached later. If during the
input operation of motion video and audio the length of
the data is known a priori, the MDBMS may have
problems choosing the proper server or disk.

• The output (play) reads the raw data from the database
according to the registering data. Hence, for decoding a
JPEG coded image, the Huffman table can be transmitted

to the decoder in advance. The transmission of the raw data follows.

• Modification usually considers the raw data. The modification of image should be done by an editor. For notion video, cutting with in (out fading usually needed. For audio data, in addition to in/out-fading, the volume, bass, treble and eventually balance can also be modified. The modification attributes are stored in registering data. Here, the attributes are defined as time-dependent functions performed during play of data.

• Modification can also be understood as a data conversion from one format to another.

• In this case, the registering data must be modified together with raw data. Another variant of modification is transformation form one medium to another, such as text-to-speech transformation.

• The conversion function, analogous to an editor, should be implemented outside the MDBMS. Such a transformauonis implemented through reading of the data,, externally 0 into another medium and encoding the transformed data in the database.

• During the delete operation the consistency of the data must be preserved, i.e., if raw data of an entry are deleted, all other data types related to raw data are deleted.

• Many queries to the MDBMS consist of a search and retrieval of the stored data. These queries are base on compatriot information.

• Here, individual patterns in the particular medium are compared with the stored raw data.

• This kind of research is not very successful. Another approach uses pattern recognition where a pattern from raw data may be stored as registering data and a comparison is based on this pattern.

DEPARTMENT OF INFORMATION TECHNOLOGY IF-364

JAYAM COLLEGE OF ENGINEER• ING AND TEHNOLOGY 23

• The current efficiency of this approach is low for MDBMSS and is only used for certain applications.

• A comparison can also be based on the 0 format of the descriptive data. Here, each audio sequence can be identified cording to its ambigu0us name and the creation time.

• Other comparisons are based on content oriented descriptive data.

• For example, the user enters the nominal phrase with a limited set of words. The MDBMS converts the input into predicates. In this case, synonyms can be used and are managed by the system.

• This concept allows for a content search, which is used for images and can be ported without any difficulties to all types of media.

• The goal of evaluating the raw and registering data is to generate the descriptive data. For example, during the storage of facsimile documents, Optical Charactet' Recognition (OCR) can be used. Otherwise, in most case, an explicit user input is required.

v CASE STUDY

• Some interesting approaches to. multimedia synchronization are described in this section and c1a according to the reference model presented previously.

• In particular, we analyze synchronization aspects in standards of multimedia information exchange and the respective run- time environments and prototype multimedia systems which comprise several layers of the synchronization reference model.

v Synchronization in MHEG

• The generic space inMHEG provides a virtual coordinate system that is used to specify the layout and relation of content objects in space and time according to the virtual axes-based specification method.

• The generic space has one time axis of infinite length measured in Generic Time Units (OTUs).

• The MHEG run-time environment must map the GTUs to Physical Tinge Units (PTUs). If no mapping is specified, the default is one GTU mapped to one millisecond.

• The presentation of content objects is based on the exchange of action objects sent to an object. Examples of actions are prepare to set the object in a presefitable

DEPARTMENT OF INFORMATION TECHNOLOGY IF-364

JAYAM COLLEGE OF ENGINEER• ING AND TEHNOLOGY 24

state, run to start the presentation and stop to end the presentation.

• Action objects can be combined to form an action list. Parallel action lists are executed in parallel. Each list is composed of a delay followed by delayed sequential actions

v MUEG Engine

• At the European networking Center in Heidelberg, an MHEG engine has been developed. The MTIEG engine is an implementation of the object layer.

• The Generic Presentation Services of the engine provide abstractions from the presentation modules used to

present the content objects. The Audio/Video-
Subsystem is a stream layer implementation.

• This component is responsible for the presentation of the continuous media streams, e.g., audio/video streams.

• The User Interface Services provide the presentation of time-independent media, like text and graphics, and the processing of user interactions, e.g., buttons and forms.

• The MHEG engine receives the MHEG objects from the application. The Object Manager manages these objects in the run-time environment.

• The interpreter processes the action objects and events. It is responsible for initiating the preparation and presentation of the objects. The Link Processor monitors the states of objects and trigger links, if the trigger conditions are fulfilled.

DEPARTMENT OF INFORMATION TECHNOLOGY IF-364

JAYAM COLLEGE OF ENGINEER• ING AND TEHNOLOGY 25

• The run-time system communicates with the presentation services by events. The User Interface Services provide events that indicate actions.

• The Audio/ Video-Subsystem provides events about the status of the presentation streams, like end of the presentation of a stream or reaching a cue point in a stream.The architecture of the MHEG engine is shown in figure