CSC 214
FILE PROCESSING

Course Development Team

Content Authoring

Sadiku P. O.

Content Editor

Mrs. Bankole

Instructional Designer

Miss. Damilola Adesodun, Mr. Olawale S. Koledafe &

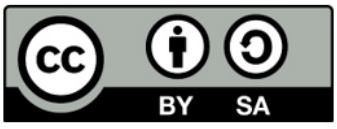
Samuel Adekanye

Graphic Designer

Bello J. B.

Published by the Centre for Open and Distance Learning,
University of Ilorin, Nigeria

E-mail: codl@unilorin.edu.ng
Website: https://codl.unilorin.edu.ng



From the Vice Chancellor

Courseware development for instructional use by the Centre for Open and Distance Learning (CODL) has been achieved through the dedication of authors and the team involved in quality assurance based on the core values of the University of Ilorin. The availability, relevance and use of the courseware cannot be timelier than now that the whole world has to bring online education to the front burner. A necessary equipping for addressing some of the weaknesses of regular classroom teaching and learning has thus been achieved in this effort.

This basic course material is available in different electronic modes to ease access and use for the students. They are available on the University's website for download to students and others who have interest in learning from the contents. This is UNILORIN CODL's way of extending knowledge and promoting skills acquisition as open source to those who are interested. As expected, graduates of the University of Ilorin are equipped with requisite skills and competencies for excellence in life. That same expectation applies to all users of these learning materials.

Needless to say, that availability and delivery of the courseware to achieve expected CODL goals are of essence. Ultimate attention is paid to quality and excellence in these complementary processes of teaching and learning. Students are confident that they have the best available to them in every sense.

It is hoped that students will make the best use of these valuable course materials.

Professor S. A. Abdulkareem
Vice Chancellor

Forward

Courseware remains the nerve centre of Open and Distance Learning. Whereas some institutions and tutors depend entirely on Open Educational Resources (OER), CODL at the University of Ilorin considers it necessary to develop its own materials. Rich as OERs are and widely as they are deployed for supporting online education, adding to them in content and quality by individuals and institutions guarantees progress. Doing it in-house as we have done at the University of Ilorin has brought the best out of the Course Development Team across Faculties in the University. Credit must be given to the team for prompt completion and delivery of assigned tasks in spite of their very busy schedules.

The development of the courseware is similar in many ways to the experience of a pregnant woman eagerly looking forward to the D-day when she will put to bed. It is customary that families waiting for the arrival of a new baby usually do so with high hopes. This is the apt description of the eagerness of the University of Ilorin in seeing that the centre for open and distance learning [CODL] takes off.

The Vice-Chancellor, Prof. Sulyman Age Abdulkareem, deserves every accolade for committing huge financial and material resources to the centre. This commitment, no doubt, boosted the efforts of the team. Careful attention to quality standards, ODL compliance and UNILORIN CODL House Style brought the best out from the course development team. Responses to quality assurance with respect to writing, subject matter content, language and instructional design by authors, reviewers, editors and designers, though painstaking, have yielded the course materials now made available primarily to CODL students as open resources.

Aiming at a parity of standards and esteem with regular university programmes is usually an expectation from students on open and distance education programmes. The reason being that stakeholders hold the view that graduates of face-to-face teaching and learning are superior to those exposed to online education. CODL has the dual-mode mandate. This implies a combination of face-to-face with open and distance education. It is in the light of this that our centre has developed its courseware to combine the strength of both modes to bring out the best from the students. CODL students, other categories of students of the University of Ilorin and similar institutions will find the courseware to be their most dependable companion for the acquisition of knowledge, skills and competences in their respective courses and programmes.

Activities, assessments, assignments, exercises, reports, discussions and projects amongst others at various points in the courseware are targeted at achieving the objectives of teaching and learning. The courseware is interactive and directly points the attention of students and users to key issues helpful to their particular learning. Students' understanding has been viewed as a necessary ingredient at every point. Each course has also been broken into modules and their component units in sequential order.

At this juncture, I must commend past directors of this great centre for their painstaking efforts at ensuring that it sees the light of the day. Prof. M. O. Yusuf, Prof. A. A. Fajonyomi and Prof. H. O. Owolabi shall always be remembered for doing their best during their respective tenures. May God continually be pleased with them, Aameen.

Bashiru, A. Omipidan
Director, CODL

Course Guide

Our world today revolve around use of data. We all needs data everyday to help us make different life decision. Therefore, there is a need to process data at a faster rate. It may intrest you to note that the manual method of processing data is slow and comes with diverse challenges such as storage, processing time and duplication.

With the advent of computer systems, challenges of traditional data processing method has been optimally addressed. Therefore, this course will introduce you to the different types Computer files together with the way they are Organized, structured, manipulated and stored in the computer memory.

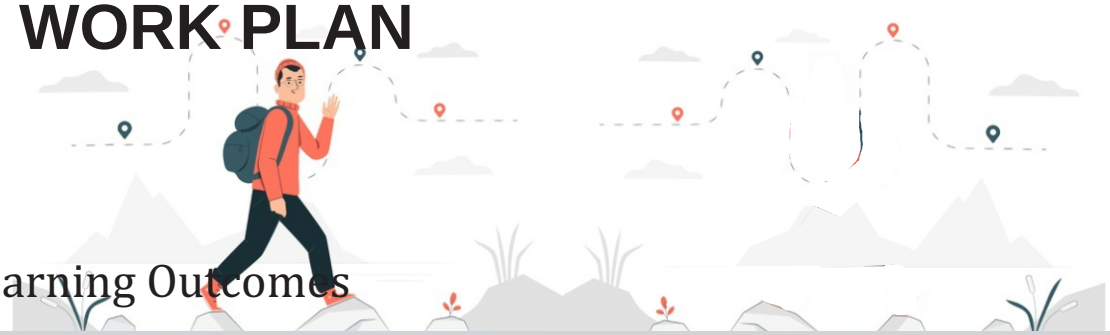
Furthermore, this course will introduce you to the Computer filing system concept since no data processing in the computer is possible without the use of files. Computer file processing and its manipulation will be discussed extensively

For your information, this course is made up of four modules and fourteen units. Each of the units explain in details the components of the computer file system. Module 1 will introduce you to data management files, while Module 2 & Module 3, will address the Input and Output systems and file allocation methods respectively. Finally in Module 4 I will extensively discuss data management facilities, file management system and file system architecture.

Course Goal

The purpose of this course is to introduce you to the practice of computer filing system. This course will equip you with skills needed as a data analyst and database administrator to meet up with the high demand for data and its processing.

WORK PLAN



Learning Outcomes

At the end of this course, you should be able to:

I.Explain the computer filing system and as well differentiate between the manual and computer filing system.

ii.State and explain the terminologies used in Computer Filing System.

iii. Explain in details, the types of data file.

iv. Identify files by their name and extension.

v.Describe the File Organization methods.

vi.Describe Job Control Language and identify Job Control Statements.

Week 01 | Week 02 | Week 03

Course Content

Module 1

Basic Concepts of Files

Unit 1: Introduction to Data Management Files

Unit 2: Basic Terminologies in File system

Unit 3: Data Files

Unit 4: File Naming

Unit 5: File Organization

Module 2

Job Control language application and Input/Output Systems Architecture

Unit 1 - Job Control Language (JCL)

Unit 2 - Structure of Job Control Language (JCL)

Unit 3 - Input/Output architecture

Related Courses

Pre-requisites for this course:

CSC 111
Introduction to Computer Science I

CSC 112
Introduction to Computer Science II

vii. Write a simple JCL Statement.
viii. Analyze the different modes involved in the input/output System

ix. State the different File Directory Structures and outline the operations carried out on a File System.
x. Categorize File Allocation methods.

xi. Analyze disasters that poses threat to computer file system and recovery strategies for lost file.
xii. Describe File Management System and its Architecture.

Week 04 | Week 05 | Week 06

Course Content

Module 1

Basic Concepts of Files

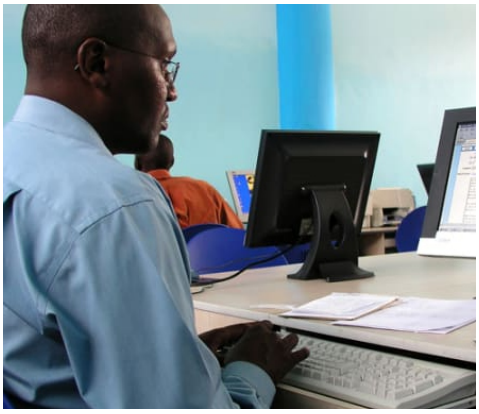
Unit 1: Introduction to Data Management Files

Unit 2: Basic Terminologies in File system

Unit 3: Data Files

Unit 4: File Naming

Unit 5: File Organization



Module 2

Job Control language application and Input/Output Systems Architecture

Unit 1 - Job Control Language (JCL)

Unit 2 - Structure of Job Control Language (JCL)

Unit 3 - Input/Output architecture

Course Requirements

Requirements for success

The CODL Programme is designed for learners who are absent from the lecturer in time and space. Therefore, you should refer to your Student Handbook, available on the website and in hard copy form, to get information on the procedure of distance/e-learning. You can contact the CODL helpdesk which is available 24/7 for every of your enquiry.

Visit CODL virtual classroom on http://codllms.unilorin.edu.ng. Then, log in with your credentials and click on CSC 214. Download and read through the unit of instruction for each week before the scheduled time of interaction with the course tutor/facilitator. You should also download and watch the relevant video and listen to the podcast so that you will understand and follow the course facilitator.

At the scheduled time, you are expected to log in to the classroom for interaction. Self-assessment component of the courseware is available as exercises to help you learn and master the content you have gone through.

You are to answer the Tutor Marked Assignment (TMA) for each unit and submit for assessment

Embedded Support Devices

Requirements for success

Throughout your interaction with this course material, you will notice some set of icons used for easier navigation of this course materials. We advise that you familiarize yourself with each of these icons as they will help you in no small ways in achieving success and easy completion of this course. Find in the table below, the complete icon set and their meaning.

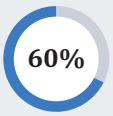
 **Introduction**	 **Learning Outcomes**	 **Main Content**

 Summary	 Tutor Marked Assignment	 Self Assessment
 Web Resources	 Downloadable Resources	 Discuss with Colleagues
 References	 Futher Reading	 Self Exploration

Grading and Assessment

TMA

CA



Exam

Total

Photo: Office space

Source: Unsplash

Module 1
Basic Concepts of Files

Photo by Pixabay from Pexel

UNIT 1

Introduction to Data Management Files

Introduction

I welcome you to the first unit of this course. In this unit, I will introduce you to the methods of storage and retrieval of documents manually, which eventually led to the computer filing system. Different types of computer files are also highlighted with its uses and management. So you must have had a basic knowledge about the computer and its hardware and software components.

Learning Outcomes

At the end of this unit, you should be able to:

- differenciate between a manual and Computer Filing system;
- give examples of the manual filing system;
- state the layers of the Computer Filing system; and
- state the types of Computer filing system.

Main Content

Introduction to Data Management Files

02mins [SAQ1]

The method of storage and retrieval of paper documents was often described as the File System. Computers inherited files together with its processing from these previous manual data processing systems. Data are organized at different levels for easy storage and retrieval in these systems. For example, in an organization, files are kept and arranged into cabinets following some particular order. This facilitates easy storage and retrieval of documents.

Tips

Do you know that before the advent of the computer, data was processed by manual methods?

A good example of a file is a list of records of students in a class in an institution. In this case, all files have a name, number or title, which distinguish it from other files. The following are examples of file:

- (a) Employee records in an organization
- (b) A cash book used in the supermarket
- (c) A price list of items in a supermarket
- (d) Student record in an institution

Therefore, a Computer file is referred to as a file that contains data, where data denotes program instruction. These files can also reside on secondary storage such as magnetic tapes, floppy disk, flash drives. Some files are processed at regular intervals for providing information, for example, student file. Others will hold data, which are required for processing without a specified or regular interval, for example, price list in a supermarket.

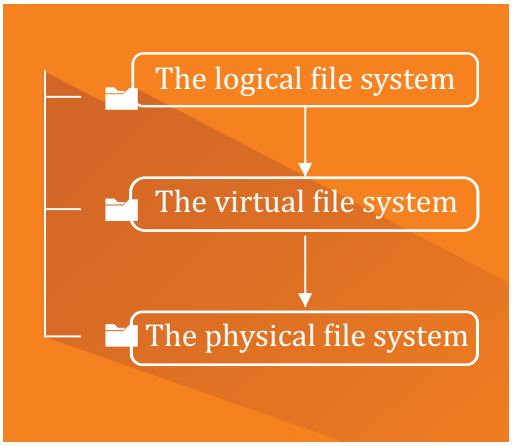
Computer File System

(2Mins) [SAQ2]

Let me tell you that in a Computer, a file system is a way in which files are named and where they are placed logically for storage and retrieval. Without a file system, stored information would not be isolated into individual files and would be difficult to identify and retrieve. As data capacities increase, the organization and accessibility of individual files are becoming even more important in data storage.

File systems can differ between operating systems(OS) such as Microsoft Windows, macOS and Linux-based systems.

A file system stores and organizes data and can be thought of as a type of index for all data contained in a storage device. These devices can include hard drivers, optical drives and flash drives.



Do you know that a computer file system consists of three layers? Sometimes the functions are combined, and sometimes the layers are explicitly separated.

The first layer, which is the logical file system, is responsible for interaction with the user application.

It provides the application program interface (API) for file operations such as OPEN, CLOSE, READ, etc., and passes the operation requested to the next layer below it for processing.

This layer also manages open file table entries and per-process file descriptors. It provides file access, directory operations, security and protection.

The second optional layer is the virtual file system. This interface allows support for multiple concurrent instances of physical file systems, each of which is called a file system implementation.

The third layer is the physical file system. This layer is concerned with the physical operation of the storage device (e.g. disk). Physical blocks being read or written are processed by this layer. It handles buffering and memory management and is responsible for the physical placement of blocks in specific locations on the storage medium. This layer interacts with the device drivers or with the channel to drive the storage device.

Uses of Computer File System

(1Mins)[SAQ3]

What are the uses of computer file system to us, how does it help us in the storage and retrieval of documents? Let'see few of these uses below:

1. Allocation of space is done by File systems in a granular manner, on multiple physical units on the device.
2. It is responsible for organizing files and directories and keeping track of which areas of the media belong to which file and which are not being used.
3. The file system manages access to both the content and the metadata about those files.

1. Disk File Systems:

A disk file system takes advantages of the ability of disk storage media to address data in a short amount of time randomly. This permits multiple users (or processes) access to various data on the disk without regard to the sequential location of the data. Examples include FAT (FAT12, FAT16, FAT32), exFAT, NTFS, HFS and HFS+, HPFS, UFS, ext2, ext3, ext4, XFS, btrfs, ISO 9660, Files-11, Veritas File System, VMFS, ZFS, ReiserFS and UDF. Journaling file systems or versioning file systems are also examples of this

disk file systems

2) Flash File System:

The flash file system considers the special abilities, performance and restrictions of flash memory devices. Sometimes, a flash memory device can be used as an underlying storage media by a disk file system, but it is much better to use a file system specifically designed for a flash device.

3) Tape File System:

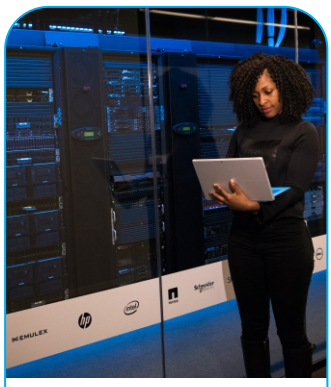
This is a file system and tape format designed to store files on tape in a self-describing form. Magnetic tapes are sequential storage media with significantly longer random data access times than disks, posing challenges to the creation and efficient management of a general-purpose file system.

This tape motion may take several seconds to several minutes to move the read/write head from one end of the tape to the other. However, writing data to a tape, erasing, or formatting a tape is often a significantly time-consuming process and can take several hours on large tapes. With many data tape technologies, it is not necessary to format the tape before over-writing new data to the tape. This is due to the inherently destructive nature of overwriting data on sequential media.

Because of the time, it can take to format a tape; typically tapes are pre-formatted so that the tape user does not need to spend time preparing each new tape for use. All that is usually

I want to notify you that in a disk file system, there is typically a master file directory and a map of used and free data regions.

Any file additions, changes, or removals require updating the directory and the used/free maps, but tape requires linear motion to wind and unwind potentially very long reels of media



A data centre Operator
Photo by: Christina Morillo
From Pexel

necessary is to write an identifying media label to the tape before use, and even this can be automatically written by the software when a new tape is used for the first time.

Database File Systems

Another concept for file management is the idea of a database-based file system. Instead of, or in addition to, hierarchical structured management, files are identified by their characteristics, like a type of file, topic, author, or similar rich metadata.

IBM DB2 for i (formerly known as DB2/400 and DB2 for i5/OS) is a database file system as part of the object-based IBM i operating system (formerly known as OS/400 and i5/OS), incorporating a single-level store and running on IBM Power Systems (formerly known as AS/400 and iSeries), designed by Frank G. Soltis who was IBM's former chief scientist for IBM i. Do you know that around 1978 to 1988, Frank G. Soltis and his team at IBM Rochester have successfully designed and applied technologies like the database file system where others like Microsoft later failed to accomplish? These technologies are informally known as 'Fortress Rochester and were in few basic aspects extended from early Mainframe technologies but in many ways more advanced from a technological perspective

Some other projects that are not "pure" database file systems but that use some aspects of a database file system:

· Many Web content management systems use a relational DBMS to store and retrieve files. For example, XHTML files are stored as XML or text fields, while image files are stored as blob fields; SQL SELECT (with optional XPath) statements retrieve the files, and allow the use of a sophisticated logic and more rich information associations than "usual file systems". Many CMSs also

have the option of storing only metadata within the database, with the standard filesystem used to store the content of files.

· Very large file systems, embodied by applications like Apache Hadoop and Google File System, use some database file system concepts.

5) Transaction File System

Some programs need to update multiple files all at once. For example, a software installation may write program binaries, libraries, and configuration files. A failure in the software installation results in an unusable program. Transaction processing introduces the isolation guarantee, which states that operations within a transaction are hidden from other threads on the system until the transaction commits, and that interfering operations on the system will be properly serialized with the transaction.

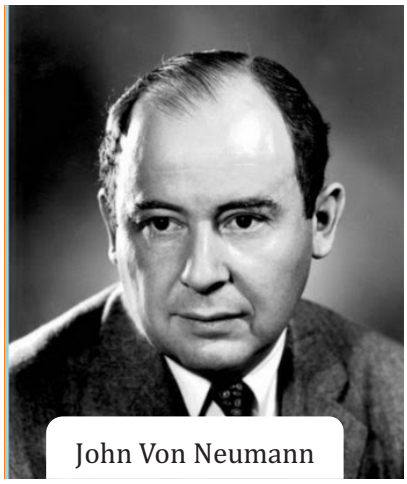
Transactions also provide the atomicity guarantee, that operations inside of a transaction are either all committed, or the transaction can be aborted, and the system discards all of its partial results. This means that if there is a crash or power failure, after recovery, the stored state will be consistent. Either the software will be completely installed, or the failed installation will be completely rolled back, but an unusable partial install will not be left on the system.

Web content management systems use a relational DBMS to store and retrieve files. For example, XHTML files are stored as XML or text fields, while image files are stored as blob fields; SQL SELECT (with optional XPath) statements retrieve the files, and allow the use of a sophisticated logic and more rich information associations than "usual file systems". Many CMSs also have the option of storing only metadata within the database, with the standard filesystem used to store the content of files.

John von Neumann was born on December 28, 1903, in Budapest, Hungary and died in Washington D. C. on February 8, 1957. He was a great mathematician with significant contribution to the theory of games and strategy, set theory and the

design of high-speed computing machines. In 1933, he was appointed one of the first six professors of the School of Mathematics in the Institute for Advanced Study at the Princeton University, USA, a position he retained until his death.

Neumann with some other people presented a paper titled "The Preliminary Discussion of the Logical Design of an Electronic Computing Instrument" popularly known as von Neumann machine. This paper contains revolutionary ideas on which the present-day computers are based. The machine has Storage, Control, Arithmetic and input/output units. The machine was to be a general-purpose computing machine. It was to be an electronic machine and introduced the concept of a stored program. The idea implied that operations in the computer were controlled by a programme stored in its memory. This program was to consist of codes that intermixed data with instructions.



John Von Neumann

As a result of this, it became possible for computations to proceed at electronic speed, perform the same set of operations or instructions repeatedly and the concept of program counter, which implied that whenever an instruction is fetched, the program counter which is a high-speed register automatically contains the address of the instruction to be executed next.

Howard Aiken of Harvard was the principal designer of the Mark I. The Harvard Mark I computer was built as a partnership between Harvard and IBM in 1944. This was the first programmable digital computer made in the U.S., but it was not a purely electronic computer. Made up of switches, relays, rotating shafts, and clutches, the machine weighed 5 tons. It was 8 feet tall, 51 feet long and incorporated 500 miles of wire. It also had a 50ft rotating shaft running its length, turned by a 5-horsepower electric motor. The Mark I ran non-stop for 15 years.

6) Network file systems:

A network file system is a file system that acts as a client for a remote file access protocol, providing access to files on a server. Programs using local interfaces can transparently create, manage and access hierarchical directories and files in remote network-connected computers. Examples of network file systems

include clients for the NFS, AFS, SMB protocols, and file-system-like clients for FTP and WebDAV.a

7) Shared Disk File System

Same external disk subsystem (usually a SAN). The file system arbitrates access to that subsystem, preventing write collisions. Examples include GFS2 from Red Hat, GPFS from IBM, SFS from DataPlow, CXFS from SGI and StorNext from Quantum Corporation.

8) Special File System

A special file system presents non-file elements of an operating system as files so they can be acted on using file system APIs. This is most commonly done in Unix-like operating systems, but devices are given file names in some non-Unix-like operating systems as well.

9) Minimal File System / Audio Cassette Storage

May I interest you to know that in the 1970s, disk and digital tape devices were too expensive for some early microcomputer users. An inexpensive basic data storage system was devised that used common audio cassette tape.

When the system needed to write data, the user was notified to press "RECORD" on the cassette recorder, then press "RETURN" on the keyboard to notify the system that the cassette recorder was recording. The system wrote a sound to provide time synchronization, then modulated sounds that encoded a prefix, the data, a checksum and a suffix. When the system needed to read data, the user was instructed to press "PLAY" on the cassette recorder. The system would listen to the sounds on the tape waiting until a burst of sound could be recognized as the synchronization. The system would then interpret subsequent sounds as data. When the data read was complete, the system would notify the user to press "STOP" on the cassette recorder. It was primitive, but it worked (a lot of the time). Data was stored sequentially, usually in an unnamed format, although some systems (such as the

Commodore PET series of computers) did allow the files to be named. Multiple sets of data could be written and located by fast-forwarding the tape and observing at the tape counter to find the approximate start of the next data region on the tape. The user might have to listen to the sounds to find the right spot to begin playing the next data region. Some implementations even included audible sounds interspersed with the data

10) Flat File Systems

Do you know that when floppy disk media was first available, this type of file system was adequate due to the relatively small amount of data space available? CP/M machines featured a flat-file system, where files could be assigned to one of 16 user areas, and generic file operations narrowed to work on one instead of defaulting to work on all of them. These user areas were no more than special attributes associated with the files; that is, it was not necessary to define a specific quota for each of these areas and files could be added to groups for as long as there was still free storage space on the disk. The early Apple Macintosh also featured a flat-file system, the Macintosh File System. It was unusual in that the file management program (Macintosh Finder) created the illusion of a partially hierarchical filing system on top of EMFS. This structure required every file to have a unique name, even if it appeared to be in a separate folder. IBM DOS/360 and OS/360 store entries for all files on a disk pack (volume) in a directory on the pack called a Volume Table of Contents (VTOC).

Put at the back of your mind that in a flat-file system, there are no subdirectories; directory entries for all files are stored in a single directory.

While simple, flat file systems become awkward as the number of files grows and makes it difficult to organize data into related groups of files. Moreso, a recent addition to the flat file system family is Amazon's S3, a remote storage service, which is intentionally simplistic to allow users the ability to customize how their data is stored. The only constructs are buckets (imagine a disk drive of unlimited size) and objects (similar, but not identical to the standard concept of a file). Advanced file management is allowed by being able to use nearly any character (including '/') in the object's name, and the ability to select subsets of the bucket's content based on identical prefixes.

Window File Systems

Let me tell you that the Microsoft Windows operating systems have always supported, and still do support, various versions of the FAT (File Allocation Table) file system. In addition to FAT, all Microsoft Windows operating systems since Windows NT support a newer file system called NTFS (New Technology File System).

All modern versions of Windows also support exFAT, a file system designed for flash drives. A file system is set up on a drive during a format.

Some operating systems other than Windows also take advantage of FAT and NTFS, but many different kinds of file systems exist, like HFS+ used in Apple product like iOS and macOS. Wikipedia has a comprehensive list of file systems if you're interested in finding out. Sometimes, the term "file system" is used in the context of partitions. For example, saying "there are two file systems on my hard drive" doesn't mean that the drive is split between NTFS and FAT, but that there are two separate partitions that are using the file system.

Do you know that most applications you come into contact with require a file system to work? so every partition should have one. Also, programs are file system-dependent, meaning you can not use a program on Windows if it was built for use in macOS.

Summary

In this unit, you have been introduced:

to the basic concepts of the File system. You have seen that the computer file system is made up of three layers namely: Logical layer, Physical Layer and the Virtual layer. The types of file system include Disk files, Transaction files, Database files, flash files and tape files.

Self-Assessment Questions

1. Give differences between a manual and Computer Filing system
2. Mention three limitations of the manual filing system
3. State the layers of the computer filing system
4. List the types of Computer filing system, you know.

Tutor Marked Assessment

- What is the third layer of the file system?
- Give the main difference between a DISK and a DISC.

References

Stenzel, N. (n.d): Basic File Management and organization. The University of Maryland Extension. Retrieved on from http//extension.umd.edu

Further Reading

- File system. Retrieved on from https://en.m.wikipedia.org
  www.cs.unc.edu~dewan242s07notesfile
- IBM Data Processing Technique ebook C20-1638-1

UNIT 2

Basic Terminologies in File System

Introduction

In this unit, I will explain the important and basic terminologies that would always be used in the file system. This detailed description also gives the classification of the file into different categories. Therefore, you should have learnt about Manual and Computer file system from the previous unit to give a basis for this study.

Learning Outcomes

At the end of this unit, you should be able to:

- define each of the terminologies in the file system;
- state the main classifications of File; and
- state which is largest among File, Record and Field.

Main Content

Photo by: **Laura-Davidson** on Unsplash

Basic Terminologies in File System

01mins [SAQ1,3]

1. Data:

Raw facts has little meaning unless they have been organized in some logical manner. It can also be defined as an item or group of items that do not make meaning on their own. Therefore, it is good to note that the smallest piece of data that can be recognized by the computer is a single character, e.g. 26 alphabets (A-Z), 10 digits (0-9), and 13 special characters including + - ? etc.

A single character requires 1-byte of computer storage.

2. Field:

This is a character or group of characters (Alphabetic or numerical) with a specific meaning. A field may be defined as a telephone number, a date of birth, customers name etc.

3. Record:

When one or more fields describe a person, place or thing, then it is said to be a Record. e.g. a student examination record, the number of courses taken etc.

4. File:

This is a collection of related or similar records. For example, a file containing the record of students currently enrolled at the University of Ilorin.

Basic Classification of Files

02mins [SAQ2]

We have different way of classifying files. Below is the classification of files together with the set of terms developed for each

1. **Physical file:** this describes how data is stored and retrieved from a storage device. The set of terms developed for its functions are:

(a) Physical record

(b) Field

(c) Character

2. **Logical files:** this describes the relationship between data. The set of terms developed for its description are

- Logical records of an entity
- The data item "attributes of an entity"

Thus a "logical file" is a file viewed in terms of what data items its records contain and what processing operations may be performed upon the file.

Therefore in general, a Physical File refers to the way the file is stored on the hardware (e.g on a disk it would be stored in blocks, which are made up of sectors) whereas a logical file is how the programmer structures the file (e.g in a series of records).

A return character can be used to illustrate the difference. To the hardware, it simply means a binary code that needs to be stored and to the programmer, it might mean "move down a line and back to the beginning of the line" in a text file.

Database Management System

05mins

May I interest you to know that a database in the most basic sense is an electronic storage depot. More precisely it may be defined as a structure that contains **entity** sets and **relationships** between those entity sets. An **entity set** is a person, place, thing or event about which you want to keep facts known as **data**. Each entity has identifying characteristics known as attributes. For example, an employee's entity attributes will likely be such things as **last name**, **first name**, **employment no**, etc.

An entity set is a collection of related entities which are named. The **Relationships** between entity sets reflect the fact that, within a database, many entities are likely to interact with other entities.

Therefore, Database management system (DBMS) is a collection of programs that enables storage, modification and extraction of information from a database.

The DBMS provides users and programmers with a systematic way to create, retrieve, update and manage data.

The DBMS manages the transactions that take place in a database, it does this by creating a **data dictionary**, which is a structure that stores **metadata** i.e. data about data. It takes care of all data storage details.

Popular DBMS are; MS Access, MySQL, and ORACLE

Now, the DBMS manages both the data and the relationships stored in the database.

Now, a DBMS makes it possible for end-users to create, read, update and delete data in a database. The DBMS essentially serves as an interface between the database and end-users or application programs, ensuring that data is consistently organized and remains easily accessible.

We have many different types of Database Management Systems ranging from small systems that run on personal computers to huge systems that run on mainframes. Example of this database systems are:

A. Computerized library systems
B. Automated teller machines
C. Flight reservation systems
D. Computerized parts inventory systems

Generally, the DBMS manages three important things: the data, the database engine that allows data to be accessed, locked and modified -- and the database schema, which defines the database's logical structure. These three foundational elements help provide concurrency, security, data integrity and uniform administration procedures. Typical database administration tasks supported by the DBMS include change management, performance monitoring/tuning and back-up and recovery. Many database management systems are also responsible for automated rollbacks, restarts and recovery as well as the logging and auditing of activity

Functions of DBMS

i. Transforms your logical data request to the match the physical data structures
ii. Allows you to create **validation rules** governing your data and then enforces those rules for you.
iii. Creates data access security through the use of a password and other devices Promotes data physical security through back-up and recovery
iv. Performs various data structure translations to let you use data from other databases.
v. Produces the complex structures that allow multiple user data access.

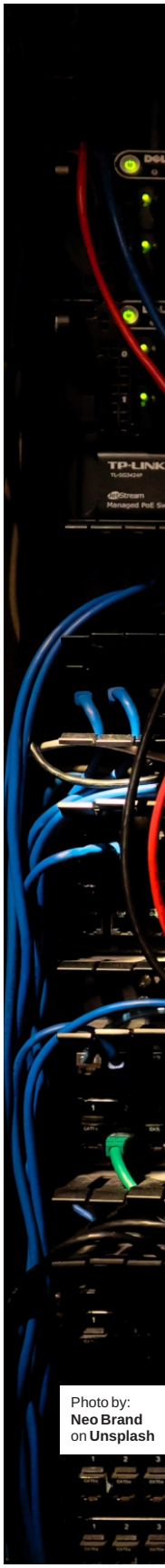


Photo by: **Neo Brand** on **Unsplash**

Classification of Relationships in Databases

It is imperative you know that in database design, it is required that relationships among entities sets be defined. Generally, relationships are classified into

1. **one-to-one relationship** written as **1:1**, this exists if an entity set A can be matched to only one entity in set B. for example, an employee manages one department, and one department is managed by only one employee.

2. **One-to-many relationship** written as **(1:M)** exists if an entity set A can be matched to only one entity in set B, but an entity in B can be matched to several entities in set A. example, an employee is employed in only one department, but a department employs many employees

3. **Many-to-many relationship(M: M)** this exits if an entity in an entity set A can be matched to many entities in entity set B and vice versa. An example is A student takes many courses, and each course is taken by several students. THE DBMS

DBMS Architecture

The architecture of a DBMS can be seen as either a single-tier or multi-tier. Enter architecture divides the whole system into related but independent n modules, which can be independently modified, altered, changed, or replaced.

Have at the back of your mind that the design of a DBMS depends on its architecture. It can be centralized or decentralized or hierarchical

Moreso, in 1-tier architecture, the DBMS is the only entity where the user directly sits on the DBMS and uses it. Any changes done here will directly be done on the DBMS itself. It does not provide handy tools for end-users. Database designers and programmers normally prefer to use single-tier architecture.

If the architecture of DBMS is 2-tier, then it must have an application through which the DBMS can be accessed. Programmers use 2-tier architecture where they access the DBMS using an application. Here the application tier is entirely independent of the database in terms of operation, design, and programming.

3-tier Architecture

A 3-tier architecture separates its tiers from each other based on the complexity of the users and how they use the data present in the database. It is the most widely used architecture to design a DBMS.

Database (Data) Tier

–At this tier, the database resides along with its query processing languages. We also have the relations that define the data and their constraints at this level.

Application (Middle) Tier

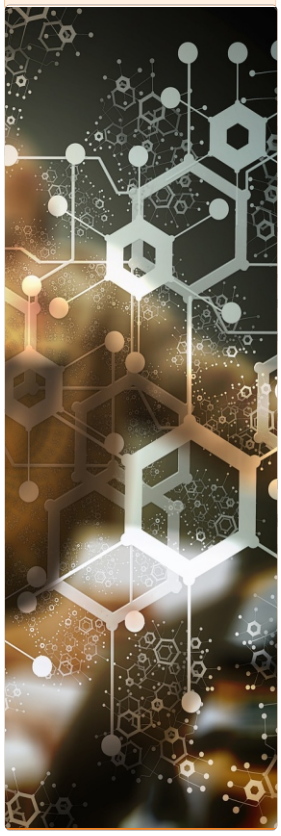
–At this tier resides the application server and the programs that access the database. For a user, this application tier presents an abstracted view of the database. End-users are unaware of any existence of the database beyond the application. You should also note that at the other end, the database tier is not aware of any other user beyond the application tier. Hence, the application layer sits in the middle and acts as a mediator between the end-user and the database.

User (Presentation) Tier

–End-users operate on this tier and they know nothing about any existence of the database beyond this layer. At this layer, multiple views of the database can be provided by the application. All views are generated by applications that reside in the application tier.

From a technical standpoint, DBMSs can differ widely. The following are terms used by DBMS to organize its information internally.

- relational
- network
- flat
- hierarchical



The internal organization can affect how quickly and flexibly you can extract information

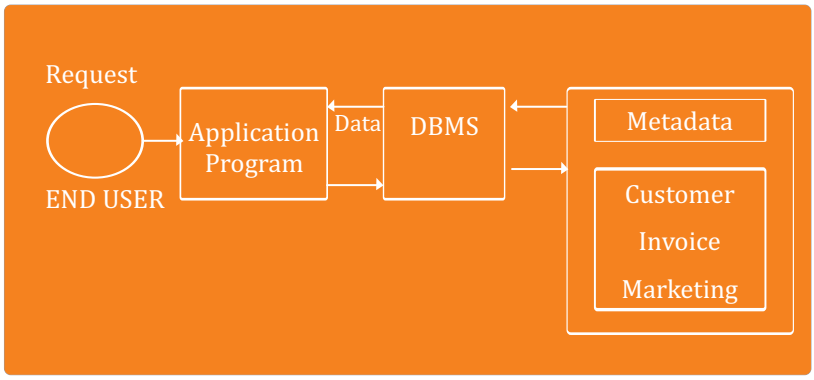


Figure 1. The DBMS managing interaction between the end-user and the database.

File System Data Management

In some high-level languages such as COBOL, BASIC, FORTRAN etc. because the file system differs from the way computer stores the data on the disk, the programmer must be familiar with the physical file structure. Therefore, every file referenced in a program requires the programmer to use complex coding to match the data characteristics and to define the precise access path to the various files and system component, this could lead to system malfunction. However as the number of files in a system expands, the systems administration becomes difficult too. Each file must have its own management system; composed of programs that allow the user to:

Let me say that in file system data management, the retrieval tasks require an extensive programming

a. Create a file structure.
b. Add data to the file.
c. Modify the data contained in the file
d. Delete data from the file.
e. List the file content.

Summary

In this unit, you have learnt the:

basic terminologies of Computer file system such as data, record and File together with file attributes and classifications. Each file must have its own management system composed of programs that allow the user to: Create the file structure, add data to the file, modify the data contained in the file and Delete data from the file

Self-Assessment Questions

1. Define the following:
   i. File
   i. Record
   ii. Field
2. Explain the two classifications of computer files
3. In terms of size, which is the largest among the following:
   i Field,
   ii Record, and
   iii File
4. State reasons for your answer.

Tutor Marked Assignment

- Differentiate between a field and a record
- What are the features of a computer file
- Which is the largest among File, Record and Field.

References

Stenzel, N. (n.d): Basic File Management and organization. The University of Maryland Extension. Retrieved on from http//extension.umd.edu

Further Reading

- www.cs.unc.edu~dewan242s07notesfile
- IBM Data Processing Technique ebook C20-1638-1
- https://en.wikipedia.org/wiki/File_system#Database_file_systems
- File system. Retrieved on from https://en.m.wikipedia.org

A file storage room

UNIT 3

Data Files

Introduction

You should already have a basic knowledge of the Computer file system, its classifications, attributes and various terminologies used in describing it. In this unit, I will be describing the composition, types and functions of Data files.

Learning Outcomes

At the end of this unit, you should be able to:

- give a detailed definition of a data file,
- state a difference between a master file and a summary file, and
- state which is largest among the types of a data file.

Main Content

Data Files

(1Mins)

Let me say that a data file (or data set) is a collection of related records that provides specific information about a fixed area of activity. They may be stored in media such as paper, cards, magnetic tape, paper tape, or direct access storage devices (DASD).

Composition of a Data File

(3Mins)[SAQ1]

Fields and Subfields:

Do you know that fields and subfields, the smallest elements of a data file, are composed of adjacent positions or characters that describe a unit of information? The leftmost position of a field is known as the high-order position; the rightmost is known as the low-order or units position.

Example: Let us consider the field labelled Date, which is six positions long, position 1 is called the high-order position and position 6, the low-order position.

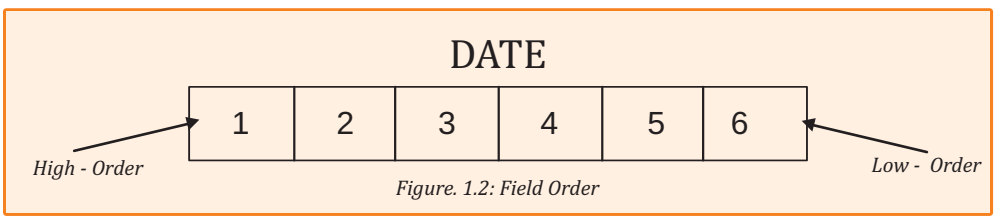


Figure. 1.2: Field Order

Subfields are meaningful subdivisions of a field to facilitate data identification and manipulation. For example, in the field called Date, the left two positions contains the subfield month, the middle two positions contain the subfield day, and the right two positions contain the subfield year.

Types of Fields and Their Functions [SAQ2]

May I interest you to know that a data recorded in a field may be classified by the function it serves.

Don't forget that control fields permit the proper identification and handling of a given record or section of a record.

1. A **record control field or key** establishes the uniqueness of the record within the file.

Example: Each data record in an employee payroll master file contains the employee number, which is the record control field. Therefore, the employee number can control the sequence of the file and the application of transaction data to the correct employee.

2. A **coding control field** identifies the record or section.

Example: Each data record in a transaction file may contain a single character field that controls the kind of information contained in the record and that indicates what effect this record will have on the master file. A digit 1 in the position may indicate that a new record is to be created in the master file with the data in the record, a digit 2 that this record contains data that will change data in a master file record, a digit 3that the master file record is to be deleted, etc.

Indicative fields usually are non-dynamic (static) and contain miscellaneous data pertinent to the record identified by the record control field. Some of these fields have a more specific nature:

3. A **statistical field** provides additional information that normally is used for the gathering of statistical information.

Example: Each employee payroll master record may have afield contained a code to indicate the sex of the employee. An M may indicate male and an F, female.

4. A **constant field** contains fixed data that otherwise might have to be developed each time the record is processed.

Example: Each employee payroll master record in a data file may have a

field containing the dollar amount of federal tax exemption allowable for that employee each week. The amount is stored in the record to avoid recalculating it each time the payroll is processed.

5. **A reference field** provides data identifying the transaction with the source document from which it was created. Reference fields are essential in providing an adequate audit trail.

Example:

In a sales application, the transaction record contains an invoice number. Should a question arise regarding the transaction, the invoice number relates the record to its source document.

6. **Quantitative fields** contain amounts and may contain sign indication. Frequently, these amounts are used in calculations, or they may be the results of computations.

Example:

Each weekly employee payroll record has a field containing hours worked. This amount field is used in calculating gross weekly earnings, which also becomes a quantitative field.

Field Characteristics

(3Mins)

We have several ways in which field can be characterized. These are:

1. **Length**

   Fields usually contain a fixed number of positions designed to hold the maximum amount of data that can occur.

2. **Class**

   The data in a field is alphabetic, numeric, or alphanumeric.

3. **Significance**

   Let me tell you that the significant digits of a numeric field are those digits which are necessary to make the numbers meaningful and which, when specified, include a fixed number of decimal positions.

Embedded blanks (blanks in the midst of significant characters)normally are unacceptable in numeric fields. The significant characters of an alphabetic or alphanumeric field include all positions from the leftmost through the rightmost non-blank character.

Embedded blanks are acceptable.

4. **Right and Left Adjustment (or Justification)**

Note that the number of significant characters for a field can vary. A right-adjusted field contains the right-most significant character in the low-order position of the field; a left-adjusted field contains the leftmost significant character in the high-order position of the field.

An alphabetic or alphanumeric field is normally left-adjusted, with the non-significant portion of the field filled with blanks. A numeric field usually is right-adjusted, with the non-significant portion of the field filled with leading zeros, although blanks may be used. Zeros are preferred to blanks because they are positive proof of the value required; blanks may represent omissions. Since blanks normally are considered of lesser numeric value than are zeros, the use of blanks instead of zeros may affect the sorting sequence. When a numeric field is left-adjusted, the non-significant portion of the field usually is filled with blanks.

Special Characteristics of Numeric Fields

Some of the special characteristics of numeric fields you must bear in mind are:

Signs. Where a numeric field may be, either positive or negative, sign coding must be present.

Decimal Positions- Usually, decimal points are assumed; this means that no space is reserved in a field for the decimal point. This conserves media storage and permits arithmetic manipulation of the field by equipment that normally does not recognize the point of a number. (See "Field CompactionTechniques" for a discussion of decimal scaling and floating decimal-point numbers.)

Recording Mode. For us to facilitate computer functions or to

reduce storage data requirements, it may be preferable for us to work with numeric data in some form other than the normal decimal notation.

Translation to and from decimal format may be accomplished by computer subroutines or by hardware capabilities of the equipment, or the data may remain in the modified mode. (See "Field Compaction Techniques" for a discussion of binary, hexadecimal, and packed format.)

Types of Data Files and Their Functions (5Mins) [SAQ3]

1. Master File

It is good to note that a master file contains the current status of a given list of items. A relatively fixed number of items are in the file over a long period of time, and the number of insertions and the number of deletions tend to be fairly well balanced despite temporary seasonal or cyclic fluctuations. Each record is subject to updating. The file is a major source of information for facilitating decisions in a particular area of operations, both internally with computer programming and externally with a management review of printed reports of data contained in the file.

Example:

An inventory master file for a hardware concern might contain one record for each item of stock. Although about 20,000 different items can be represented on the file at any given time, some items, such as lawn sprinklers, may be stocked only in the summer and others, such as snow shovels, only in the winter. The quantity on hand for a certain item must be updated to reflect any change in stock caused by such transactions as sales, returns, receipts, etc. Based on a minimum balance on hand, the computer, through programming, can determine the time to reorder. Periodically, the information on the file can be used to print out reports indicating sales trend, slow-moving items, low-profit sellers, etc., which can be reviewed by management.

2 Transaction File

You should know that transaction files keep a record of all transactions that have taken place, usually over a set period. For example, a bank machine may

keep record of all transactions in the last 24 hours

The primary purpose of a transaction file is to contain activity or inquiry records that will be used to examine and/or update a master file. Each activity record contains data about an occurrence that will affect the master file in some way. Another example is an activity such as receipts, sales, returns, etc. could be be contained in the transaction file that is used to update a master inventory file.

4. **History File**

You should also be aware that history file can be an obsolete master file or a compilation of transaction records that have affected a master file within a particular period. It is maintained primarily to gather statistical data or to capture sufficient detail of past processing to facilitate reconstruction of a master file.

5. **Summary Files**

Kindly note that a summary file represents data from another file reduced to a more concise form. The information from several records in the original file can be shown in aggregate form on one record of the summary file by using broader criteria for record uniqueness. For example, employee earnings records in a payroll master file may be summarized

into fewer records showing total earnings by the department. Depending upon its use, a summary file also may be considered as a master file, a transaction file, or a history file.

6. **Trailer File**

Be informed that a trailer file contains detail records associated with particular records in another file. The latter often are called prime records and constitute a prime file. The records in the trailer file provide additional information to augment the data found in the associated prime records. Trailer files may be processed individually or together with their prime file.

Example: Inventory Parts File

(a) For daily processing of inventory updating only the prime file is used.

(b) For the preparation of cross-reference part number and name list only the trailer file is handled.

(c) For periodic inventory status reports requiring part name as well as part number and the associated quantitative data, both the prime file and trailer files are processed simultaneously.

Summary

In this unit, I had explained that:

a data file is a collection of related records that provides specific information about a fixed area of activity. I also explained that we have different types of data files such as Master file, Trailer file, Transaction file, Summary file and History file.

Self-Assessment Questions

1. What is a Data File?
2. State three characteristics of a numeric field
3. List and briefly explain the different types of Data file you know

Tutor Marked Assessment

- Give a major difference between the Master file and every other file
- Explain the Transaction file and differentiate from the summary file

References

Stenzel, N. (n.d): Basic File Management and organization. The University of Maryland Extension. Retrieved 17th October 2019from http//extension.umd.edu

File processing.Computing Center publication 2000 p.(2)

www.cs.unc.edu~dewan242s07notesfile

Further Reading

- https://www.unf.edu › public › cop4610 › ree › Notes › PPT
- https://www.edouniversity.edu.ng › oerrepository › articles › file_organization

UNIT 4

File Naming

Introduction

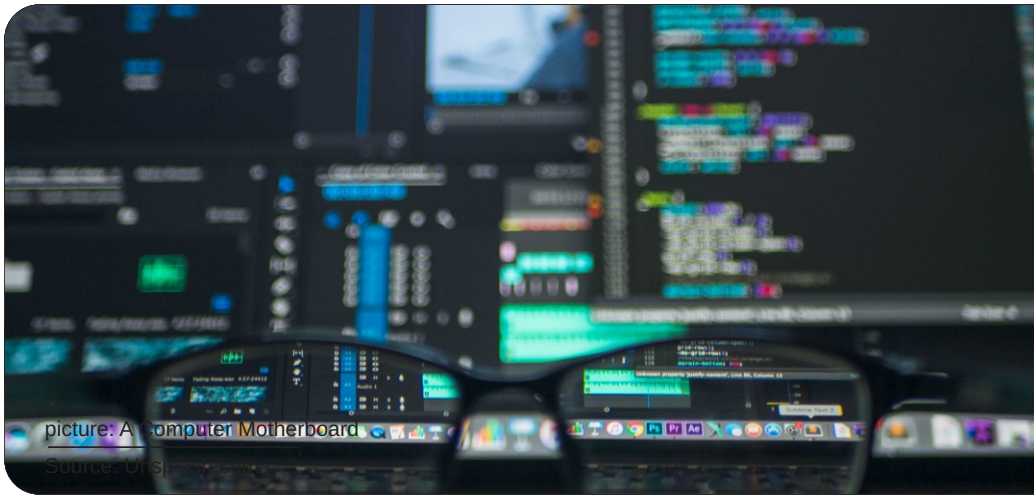
In this unit, I will be discussing the basic rules of naming a file and its extensions. We will give examples of different types together with their names and extension. Therefore, it is necessary for you to have studied the previous units where Computer file system, types and classifications have been introduced. This will give a good basis for a better understanding of this unit.

Learning Outcomes

At the end of this unit, you should be able to:

- mention some examples of file extensions,
- explain why a file extension is a suffix to the name of a computer file, and
- study the usage of File extension by Operating systems.

Main Content



picture: A Computer Motherboard
Source: Unsp

File Naming and Attributes

(3Mins)

I want you to know that files are abstraction mechanisms. They provide a way to store information and read it back later.

So, a file name is a principal identifier of a file and therefore, good file names provide useful clues to the content, status and version of a file, uniquely identify a file and help in classifying and sorting files. File names which reflect the file content also facilitate searching and discovery of these files.

File Naming Convention

Let me define File Naming Convention (FNC) as a framework for naming your files in a way that describes what they contain and how they relate to other files. Developing an FNC is done through identifying the key elements of the project, the important differences and commonalities between your files. These elements could include things like the date of creation, author's name, project name, name of a section or a sub-section of the project, the version of the file, etc. An advantage to using unique and standardized filenames is the ability to follow path names and link to other systems that require unique filenames.

You should also note that the exact rules for file naming vary somewhat from system to system, but all operating systems allow strings of one to eight letters as legal file names. The file name is chosen by the person creating it, usually to reflect its contents. There are few constraints on the format of the file name: It can comprise the letters A-Z, numbers 0-9 and special characters $ # & + @!( ) - { } ' ` _ ~ as well as space. The only symbols that cannot be used to identify a file are * | <> \ ^ =? / [ ] ' ; , plus control characters.

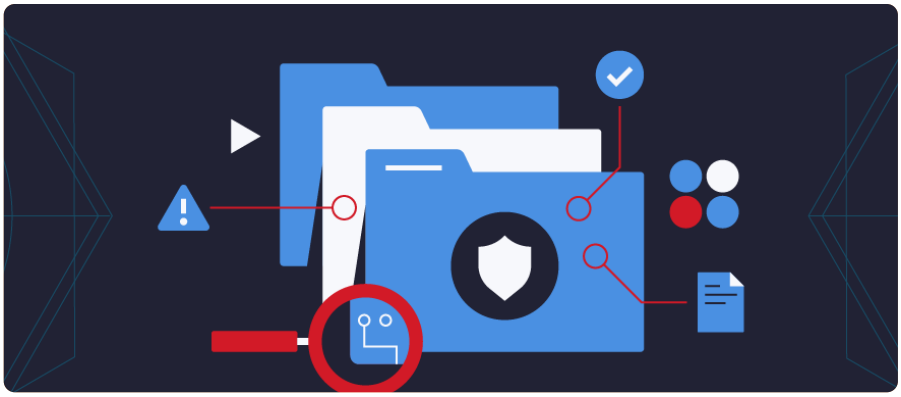
Usually, a file would have two parts with a dot (".") separating them. You should note that the part on the left side of the period character is called the **main name** while the part on the right side is called the **extension**. A good example of a file name is "csc604.doc."

Be informed that file extension differentiates between different types of files. We can have files with same names but different extensions.

It is good to note that when choosing a file name there are different rules for different operating systems that can present problems when files are moved from computer to another. For example,

Microsoft Windows is case insensitive, so files like MYEBOOKS, myebooks, MyEbooks are all the same to Microsoft Windows.

However, under the UNIX operating system, all three would be different files as, in this instance, file names are case sensitive.



Elements in a file name

Common elements that you should consider when developing a file naming strategy:

1. Version number (also see 'Data authenticity');
2. Date of creation (date format should be YYYY-MM-DD);
3. Name of creator;
4. Description of content;
5. Name of research team/department associated with the data;
6. Publication date;
7. Project number

Features of a good File Name

- File Names must be Unique
- File Names must Indicate what the file contains
- File Names must Reflect how you work with your information i.ewhat is significant, what is most likely to be searched for, who is the audience.
- File Names must be Naturally ordered
- File Names must be Consistent and understood by everyone

File Name Extension

You should note that a filename extension is a suffix to the name of a computer file applied to indicate the encoding convention or file format of its contents. In some operating systems (for example UNIX) it is optional, while in some others (such as DOS) it is a requirement. Some operating systems limit the length of the extension (such as DOS and OS/2, to three characters) while others (such as UNIX) do not. Some operating systems (for example RISC OS) do not use file extensions.

Below are several examples of file types and extensions together with the full meaning.

Executable File Extensions you should know

File Extension	File Extension Meaning
.apk	Android Package File
.bat	Batch File
.bin	Binary File
.cgi	Common Gateway Interface Script
.com	MS-DOS command file
.exe	Executable file
.jar	Java Archive File
.py	Python file
.wsf	Windows Script File

Table1.1: Executable file extension and meaning

Audio File Extensions you should know [SAQ1]

File Extension	File Extension Meaning
.aif	AIF/Audio Interchange audio file
.cda	CD audio trach file
.iff	Interchange File Format
.mid or midi	MIDI audio file
.mp3	Mp3 audio file
.mpa	MPEG-2 audio file
.wav	WAVE file
.wma	Windows Media Audio file
.wpl	Windows Media Player playlist

Table 1.2: Audio file extension and meaning

Video File Extensions

File Extension	File Extension Meaning
.avi	Audio Video Interleave File
.flv	Adobe File
.h264	H.264 Video File
.m4v	Apple Mp4 Video FIle
.mkv	Matroska Mutimedia Container
.mov	Apple QuickTime Movie File
.mp4	MPEG-4 Video File
.mpg or mpeg	MPEG Video File
.rm	Real Media File
.swf	Shockwave Flash File
.vob	DVD Video Object File
.wmv	Windows Media Video File

Table 1.3: Video file extension and meaning

Text File Extension

File Extension	File Extension Meaning
.doc and .docx	Microsoft Word File
.odt	OpenOffice Writer Document file
.msg	Outlook Writer Document file
.pdf	Portable Document Format
.rtf	Rich Text Format
.txt	Plain text file
.wpd	WordPerfect document

Table 1.4: Text file extension and meaning

Spreadsheet File Extensions

File Extension	File Extension Meaning
.ods	Ope Office Calc Spreadsheet File
.xlr	Microsoft Works Spreadsheet file
.xls	Microsoft Excel file
.xlsx	Microsoft Excel Open XML Spreadsheet file

Table 1.5: spreadsheet file extension and meaning

Presentation File Extensions

File Extension	File Extension Meaning
.key	Keynote Presentation
.odp	OpenOffice Impress Presentation file
.pps	PowerPoint Slideshow
.ppt	PowerPoint Presentation
.pptx	PowerPoint Open XML Presentation

Table 1.6: Presentation file extension and meaning

Database File Extensions

[SAQ2]

File Extension	File Extension Meaning
.accdb	Access 2007 Database File
.csv	Comma Separated Value File
.dat	Data File
.db or .dbf	Database File
.log	Log File
.mdb	Microsoft Access Database File
.pdb	Program Database
.sav	Save File (e.g game save file)

File Extension	File Extension Meaning
.sql	SQL/Structured Query Language database file
.tar	Linux/Unix arball file archive

Table 1.7: Database file extension and meaning

System Related File Extensions

File Extension	File Extension Meaning
.bak	Back-up file
.cab	Windows Cabinet file
.cfg	Configuration file
.cpl	Windows Control Panel file
.cur	Windows Cursor file
.dll	DLL file
.dmp	Dump file
.drv	Device Driver file
.icns	macOS X icon resource file
.ico	Icon file
.ini	Initialization file
.ink	Window Shortcut file
.msi	Windows Installer Package
.sys	Windows System File
.tmp	Temporary file

Table 1.8: system related file extension and meaning

Web File Extensions

File Extension	File Extension Meaning
.asp and aspx	Active Server Page file
.cer	Internal Security Certificate
.cfm	ColdFusion Markup File
.cgi or .pl	Perl Script File
.css	Cascading Style Sheet file
.htm and .html	HTML / Hypertext Markup Language file
.js	JavaScript File
.jsp	Java Server Page file
.part	Partially downloaded file
.php	PHP Source Code file
.rss	RSS / Rich Site Summary file
.xhtml	XHTML / Extensible Hypertext Markup Language file

Table 1.9: Web file extension and meaning

Image File Extensions

File Extension	File Extension Meaning
.ai	Adobe Illustrator
.bmp	Bitmap Image File
.gif	GIF/Graphical Interchange Format
.ico	Icon File
.jpeg or .jpg	JPEG image
.max	3ds Max Scene file
.obj	Wavefront 3D Object File
.png	PNG / Portable Network Graphic image

File Extension	File Extension Meaning
.ps	Partially downloaded file
.psd	PSD/ Adobe Photoshop Document image
.svg	Scalable Vector Graphics file
.tif or .tiff	TIFF Image
..3ds	3D Studio Scene
.3dm	Rhino 3D Model

Table 1.10: image file extension and meaning

Summary

So far in this unit, you have learnt that:

that files provide a way to store information and read it back later, therefore there is a need for a unique Filename which is generally chosen by the person creating it, usually to reflect its contents. It can comprise the letters A-Z, numbers 0-9 and special characters $ # & + @ ! ( ) - { } ' ` _ ~ as well as space. This is followed by a dot and then the file extension. A good example of a file name is "csc604.doc."

File extension differentiates between different types of files and therefore we can have files with same names but different extensions. Different examples of these extensions have been stated.

A computer file is also known to have different attributes like filename, location, type, size, protection, usage count

Self-Assessment Questions

1. State two examples of audio files and give the full meaning of their extension each.
2. State two files indicating their extensions and filenames.
3. Give an example of an Operating system that does not use a File extension

Tutor Marked Assessment

- State four characters that are unacceptable when naming a file
- Explain the usage count attribute of a file.

References

Folder-File-Naming-Convention-10Rules-Best-Practice

Stenzel, N. (n.d): Basic File Management and organization. University of Maryland Extension. Retrieved 17th October 2019 from http//extension.umd.edu

File naming and Organization. Retrieved from https://guides.library.upenn.edu

Further Reading

- lecture10teaching.csse.uwa.edu.auunitsCITS2230handoutsLecture10lecture10.
- https://guides.lib.purdue.edu/c.php?g=353013&p=2378293

A computer motherboard

Photo by:
Christian Wiediger on Unsplash

UNIT 5

File Organization

Introduction

In this unit, you will learn the different File organization methods such as the serial, sequential, indexed-sequential, indexed and direct organization.

Learning Outcomes

At the end of this unit, you should be able to:

- identify which of the forms of file organizations is least complicated,
- identify which file organization has the first field in each record referred to as the key field, and
- state a difference between Pile and sequential files..

Main Content

picture: A Computer Motherboard
Source: Unsplash.com

File Organization

(7Mins)

I want you to know that a file organization deals with the relationship between the control fields of a file record and the physical location of that record in the storage medium.

Types of File Organization

Although a file of records can be arranged in storage media in different ways, all of the ways can be classified by either of two main techniques. They are sequential and random technique.

Sequential

[SAQ1]

Note that sequential order implies that there is a certain order (which can either be numeric or alphanumeric, ascending or descending), of the adjacent records in the file.

Particular fields, located in the same relative positions within the fixed section of all data records of the file, are selected as sort control fields for a specific file sequence.

Random (Non-Sequential)

Also note that a random file organization contains records

stored without regard to the sequence of their record control fields. For example, a random file organization for tape master files is impractical, since the desired record can fall anywhere within the file limits, and each search for a specific record must begin with the first record of the file.

Generally, we have different File Organization methods, these are given below.

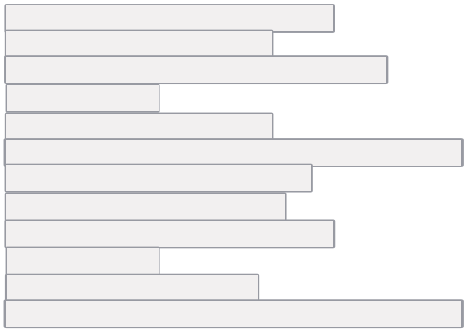
- The pile/serial
- The sequential file
- The indexed sequential file
- The indexed file
- The direct, or hashed file

1 The Pile/Serial [SAQ2]

Be informed that the least-complicated form of file organization may be termed the pile/serial. Data are collected in the order in which they arrive. Each record consists of one burst of data. The purpose of the pile/serial is simply to accumulate the mass of data and save it. Records may have different fields or similar fields in different orders. Thus, each field should be self-describing, including a field name as well as a value. The length of each field must be implicitly indicated by delimiters, explicitly included as a subfield, or known as default for that field type. Because there is no structure to the pile/serial file, record access is by exhaustive search. That is, if we wish to find a record that contains a particular field with a particular value, it is necessary to examine each record in the pile until the desired record is found or the entire file has been searched. If we wish to find all records that contain a particular field or contain that field with a particular value, then the entire file must be searched.

Pile/serial files are encountered when data are collected and stored before processing or when data are not easy to organize. This type of file uses space well when the stored data vary in size and structure; is perfectly adequate

for exhaustive searches, and is easy to update. However, beyond these limited uses, this type of file is unsuitable for most applications.



(a) Pile File

Variable-length records
Variable set of fields
Chronological order

2 The Sequential File

[SAQ2,3]

Do you know that the most common form of file structure is the sequential file? A fixed format is used for records. All records are of the same length, consisting of the same number of fixed-length fields in a particular order. Because the length and position of each field are known, only the values of fields need to be stored; the field name and length for each field are attributes of the file structure. One particular field, usually the first field in each record, is referred to as the **key field.**

The key field uniquely identifies the record; thus key values for different records are always different. Further, the records are stored in key sequence: alphabetical order for a text key, and numerical order for a numerical key.

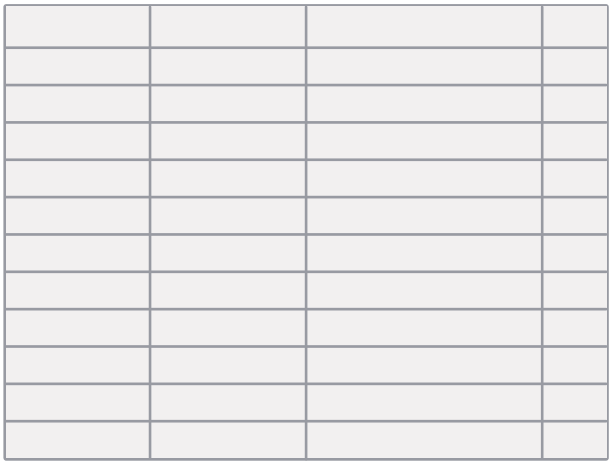
Note that sequential files are typically used in batch applications and are generally optimum for such applications if they involve the processing of all the records. The sequential file organization is the only one that is easily stored on tape as well as disk.

You should also note that for interactive applications that involve queries and/or updates of individual records, the sequential file provides poor performance.

Access requires the sequential search of the file for a key match. If the entire file, or a large portion of the file, can be brought into main memory at one time, more efficient search techniques are possible.

Additions to the file also present problems. Typically, a sequential file is stored in the simple sequential ordering of the records within blocks. That is, the physical organization of the file on tape or disk directly matches the logical organization of the file. In this case, the usual procedure is to place new records in a separate pile file, called a log file or transaction file. Periodically, a batch update is performed that merges the log file with the master file to produce a new file in a correct key sequence.

An alternative is to organize the sequential file physically as a linked list. One or more records are stored in each physical block. Each block on disk contains a pointer to the next block. The insertion of new records involves pointer manipulation but does not require that the new records occupy a particular physical block position. Thus, some added convenience is obtained at the cost of additional processing and overhead.



Fixed - length records
Fixed se of fields in fixed order
Sequential order based on key field

(a) Sequential File

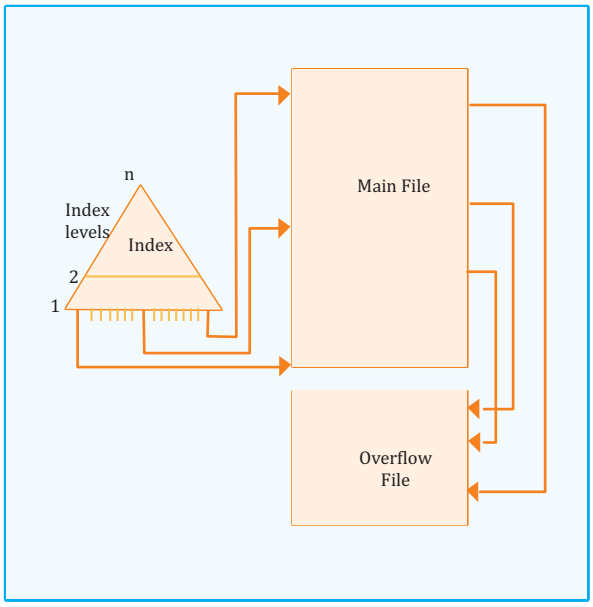
3 The Indexed Sequential File [SAQ2,3]

Be informed that in this type of file system, a popular approach to overcoming the disadvantages of the sequential file is the indexed sequential file. The indexed sequential file maintains the key characteristic of the sequential file: records are organized in sequence based on a key field. Two features are added: an index to the file to support random access, and an **overflow file.**

The index provides a lookup capability to quickly reach the vicinity of a desired record. The overflow file is similar to the log file used with a sequential file but is integrated so that a record in the overflow file is located by following a pointer from its predecessor record.

You should be aware that in the simplest indexed sequential structure, a single level of indexing is used. The index, in this case, is a simple sequential file. Each record in the index file consists of two fields: a key field, which is the same as the key field in the main file, and a pointer into the main file. To find a specific record, the index is searched to find the highest key value that is equal to or precedes the desired key value. The search continues in the main file at the location indicated by the pointer.

Also have it in mind that additions to the file are handled in the following manner: Each record in the main file contains an additional field not visible to the application, which is a pointer to the overflow file. When a new record is to be inserted into the file, it is added to the overflow file. The record in the main file that immediately precedes the new record in logical sequence is updated to contain a pointer to the new record in the overflow file. If the immediately preceding record is itself in the overflow file, then the pointer in that record is updated. As with the sequential file, the indexed sequential file is occasionally merged with the overflow file in batch mode.



The indexed sequential file greatly reduces the time required to access a single record, without sacrificing the sequential nature of the file. To process the entire file sequentially, the records of the main file are processed in sequence until a pointer to the overflow file is found, then accessing continues in the overflow file until a null pointer is encountered, at

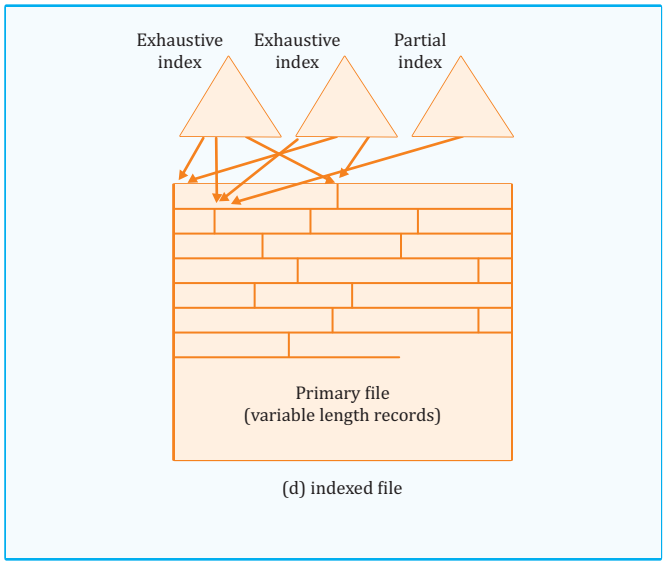
To provide even greater efficiency in access, multiple levels of indexing can be used. Thus the lowest level of the index file is treated as a sequential file and a higher-level index file is created for that file. which time accessing of the main file is resumed where it left off.

4 The Indexed File

Put at the back of your mind that the indexed sequential file retains one limitation of the sequential file: effective processing is limited to that which is based on a single field of the file. For example, when it is necessary to search for a record based on some other attributes than the key field, both forms of the sequential file are inadequate. In some applications, the flexibility of efficiently searching by various attributes is desirable.

For us to achieve this flexibility, a structure is needed that employs multiple indexes, one for each type of field that may be the subject of a search. In the general indexed file, the concept of sequentiality and a single key are abandoned. Records are accessed only through their indexes. The result is that there is now no restriction on the placement of records as long as a pointer in at least one index refers to that record. Furthermore, variable-length records can be employed.

Two types of indexes are used. An exhaustive index contains one entry for every record in the main file. The index itself is organized as a sequential file for ease of searching. A partial index contains entries to records where the field of interest exists. With variable-length records, some records will not contain all fields. When a new record is added to the main file, all of the index files must be updated. Indexed files are used mostly in applications where timeliness of information is critical and where data are rarely processed exhaustively. Examples are airline reservation systems and inventory control systems.



(d) indexed file

5 The Direct or Hashed File

Let me say that the direct or hashed file exploits the capability found on disks to access directly any block of a known address. As with sequential and indexed sequential files, a key field is required in each record. However, there is no concept of sequential ordering here. The direct file makes use of hashing on the key value. Direct files are often used where very rapid access is required, where fixed length records are used, and where records are always accessed one at a time. Examples are directories, pricing tables, schedules, and name lists.

Summary

So far in this unit, you have learnt that:

File organization deals with the relationship of the control fields of a file record to the physical location of that record in the storage medium. The different File Organization methods that have been explained are: Pile, Sequential, Indexed sequential, Indexed and Direct.

Self-Assessment Questions

1. Explain the Sequential File Organization method
2. State a difference between Pile and sequential files
3. What is a Key Field?

Tutor Marked Assessment

- Give two (2) examples of sequential file organization method

References

- Stenzel, N. (n.d): Basic File Management and organization. University of Maryland Extension. Retrieved 17th October 2019 from http//extension.umd.edu
- File naming and Organization. Retrieved from https://guides.library.upenn.edu

Further Reading

Data Processing Technique Handbook for IBM, Retrieved from http://bitsavers.trailing-edge.com › C20-1638-1_Data_File_Handbook_Mar66

Photo: A Programmer

Source:Olia Danilevich

Module 2

Job Control language application and Input/Output Systems Architecture

Photo By: David Rangel on Unsplash

UNIT 1

Job Control Language (JCL)

Introduction

Welcome to Module 2 Unit 1. In this unit, I will introduce you to Job control language, describing the common statements used and the use of the job card.

Learning Outcomes

At the end of this unit, you should be able to:

- identify Job Control Statements,
- define Job Control Language, and
- explain the use of a job card.

Main Content

Job Control Language (JCL)

04mins [SAQ1]

I want you to know that Job control language (JCL) is a scripting language executed on an IBM mainframe operating system. It consists of control statements that designate a specific job for the operating system.

You should note that JCL provides a means of communication between the application program, operating system and system hardware.

You should also note that JCL is considered to be one of the rude script languages run on IBM OS/360 batch systems. It can define data set names, parameters and system output devices.

There are **two** distinct IBM Job Control languages:

1. one for the operating system lineage that begins with DOS/360 and whose latest member is z/VSE; and
2. The other for the lineage from OS/360 to z/OS, the latter now including JES extensions, JobEntryControl Language (JECL).

You should be aware that they share some basic syntax rules and a few basic concepts but are otherwise very different.

Don't forget that one common feature in both DOS and OS JCL is the unit of work, which is called a job. A job consists of several small steps for running a specific program and is identified by cards called job cards, which indicate the beginning of the job and define exactly how the job is to be executed.

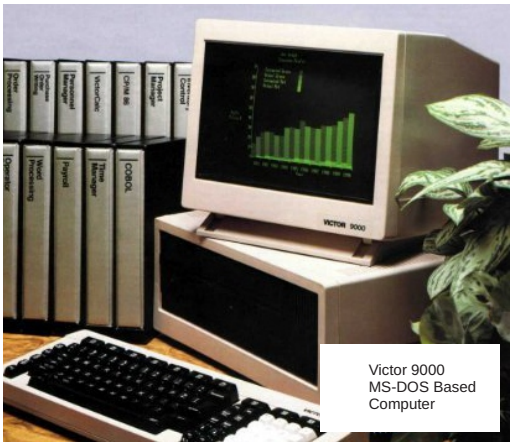
Be informed that both DOS and OS operating systems use 71 characters per line. However, the maximum length is 80 characters. Characters 73-80 are used for locating the error areas reported by the OS.

it is good to note that when a JCL statement becomes too lengthy and exceeds the 71-character limit, it can be extended using a continuation card. A statement can be continued to as many cards as necessary by ending all the JCL cards excluding the last card at an instance where a comma is used, or by using (//) at the start of the continuation card in column one and using at least a one-space character.

Basic syntax

You should be aware that both DOS and OS JCL have a maximum usable line length of 80 characters, because when DOS/360 and OS/360 were first used the main method of providing new input to a computer system was 80-column punched cards. It later became possible to submit jobs via disk or tape files with longer record lengths, but the operating system's job submission components ignored everything after character 80.

Strictly speaking both operating system families use only 71 characters per line. Characters 73-80 are usually card sequence numbers which the system printed on the end-of-job report and are useful for identifying the locations of any errors reported by the operating system. Character 72 is usually left blank, but it can contain a nonblank character to indicate that the JCL statement is continued onto the next card.



Victor 9000
MS-DOS Based
Computer

All commands, parameter names and values have to be in capitals, except for USS filenames.

Let me tell you that all lines except for in-stream input (see below) have to begin with a slash "/", and all lines which the operating system processes have to begin with two slashes // - always starting in the first column. However, there are two exceptions: the delimiter statement and the comment statement. A delimiter statements begins with a slash and an asterisk (/*), and a comment statement in OS JCL begins with a pair of slashes and asterisk (//*) or an asterisk in DOS JCL.

You should also note that many JCL statements are too long to fit within 71 characters, but can be extended on to an indefinite number of continuation cards by:

OS JCL	DOS JCL
Ending all actual JCL cards except the last at a point where the syntax requires a comma (,)	Ending all actual JCL cards except the last at a point where the syntax requires a comma (,) and a non-blank character in column 72
Starting each continuation card with // in column 1 and then at least 1 space	Starting each continuation card with spaces and continuing in column 15

Table 2.1

The structure of the most common types of card is:

OS JCL	DOS JCL
• // • Name field for this statement, following // with no space between. If this statement does not have a name at least one blank immediately follows the //.	• // (spaces if this is a continuation of a previous line) • Statement type for this statement, following // with a space between. • Space(s) • Name of resource

Table 2.2

• Space(s) • Statement type • Space(s) Parameters, which vary depending on the statement type, separated by commas and with no space between them.	• Space(s) • Parameters, which vary depending on the statement type, separated by commas and with no space between them. Positional parameters, followed by keyword parameters.

Table 2.2

In-stream input

05mins

Do you know that DOS and OS JCL both allow in-stream input? i.e. "cards" which are to be processed by the application program rather than the operating system.

Data which is to be kept for a long time will normally be stored on disk, but before the use of interactive terminals became common the only way to create and edit such disk files was by supplying the new data on cards.

Also note that DOS and OS JCL have different ways of signaling the start of in-stream input, but both end in-stream input with /* at column 1 of the card following the last in-stream data card. This makes the operating system resume processing JCL in the card following the /* card.

OS JCL: DD statements can be used to describe in-stream data, as well as data sets. A DD statement dealing with in-stream data has an asterisk (*) following the DD identifier, e.g. //SYSIN DD *. JCL statements can be included as part of in-stream data by using the DD DATA statements.

An operand named **DLM** allowed specifying a delimiter (default is "/*"). Specifying an alternate delimiter allows JCL to be read as data, for example to copy procedures to a library member or to submit a job to the *internal reader.*

An example, which submits a job to the Internal Reader **(INTRDR)** is:

```
//SUBMEXECPGM=IEBGENER
//SYSPRINT DD  SYSOUT=Z
//SYSUT2  DD  SYSOUT=(A,INTRDR)
//SYSIN   DD  DUMMY
//SYSUT1  DD  DATA,DLM=ZZ
//RUNLATRJOBACCT,MANIX,CLASS=A.TYPRUN=HOLD
//* ^ a JOB to run later
//CPUHOGEXECPGM=PICALC1K
//OUTPUT  DD  DSN=PICALC.1000DGTS,SPACE=(TRK,1),DISP=(,KEEP)
ZZ
//* ^ as specified by DLM=ZZ
//DROPOLDREXECPGM=IEFBR14
//DELETE4 DDDSN=PICALC.4DGTS,DISP=(OLD,DELETE)
//DELETE5 DDDSN=PICALC.5DGTS,DISP=(OLD,DELETE)
```

1. The program called PICALC1K will await (TYPRUN=HOLD) being released manually
2. Two files, PICALC.4DGTS & PICALC.5DGTS will be deleted NOW.

· DOS JCL: Simply enter the in-stream data after the EXEC card for the program.

Complexity

Much of the complexity of OS JCL, in particular, derives from the large number of options for specifying dataset information. While files on Unix-like operating systems are abstracted into arbitrary collections of bytes, with the details handled in large part by the operating system, datasets on OS/360 and its successors expose their file types and sizes, record types and lengths, block sizes, device-specific information like magnetic tape density, and label information.

Although there are system defaults for many options, there is still a lot to be

specified by the programmer, through a combination of JCL and information coded in the program. The more information coded in the program, the less flexible it is, since information in the program overrides anything in the JCL; thus, most information is usually supplied through JCL.

Let us take this for example, to copy a file on Unix operating system, the user would enter a command like:

```
cp oldFilenewFile
```

The following example, using JCL, might be used to copy a file on OS/360:

```
//IS198CPYJOB(IS198T30500),'COPY JOB',CLASS=L,MSGCLASS=X
//COPY01EXECPGM=IEBGENER
//SYSPRINT DD SYSOUT=*
//SYSUT1   DD DSN=OLDFILE,DISP=SHR
//SYSUT2   DD DSN=NEWFILE,
//DISP=(NEW,CATLG,DELETE),
//SPACE=(CYL,(40,5),RLSE),
//DCB=(LRECL=115,BLKSIZE=1150)
//SYSIN  DDDUMMY
```

Be aware that for both DOS and OS the unit of work is the job.

A job consists of one or several steps, each of which is a request to run one specific program. For example, before the days of relational databases, a job to produce a printed report for management might consist of the following steps: a user-written program to select the appropriate records and copy them to a temporary file; sort the temporary file into the required order, usually using a general-purpose utility; a user-written program to present the information in a way that is easy for the end-users to read and includes other useful information such as sub-totals; and a user-written program to format selected pages of the end-user information for display on a monitor or terminal.

I want you to know that in both DOS and OS JCL the first "card" must be the JOB card, which:

- identifies the job.
- usually provides information to enable the computer services department to bill the appropriate user department.
- defines how the job as a whole is to be run, e.g. its priority relative to other jobs in the queue.

Do you know that procedures (commonly called procs) are pre-written JCL for steps or groups of steps, inserted into a job? Both JCLs allow such procedures. Procs are used for repeating steps which are used several times in one job, or in several different jobs. They save programmer time and reduce the risk of errors. To run a procedure, one simply includes in the JCL file a single "card" which copies the procedure from a specified file and inserts it into the job stream. Also, procs can include parameters to customize the procedure for each use.

Originally, mainframe systems were oriented toward batch processing. Many batch jobs require setup, with specific requirements for main storage, and dedicated devices such as magnetic tapes, private disk volumes, and printers set up with special forms. JCL was developed as a means of ensuring that all required resources are available before a job is scheduled to run.

For example, many systems, such as Linux allow identification of required datasets to be specified on the command line, and therefore subject to substitution by the shell, or generated by the program at run-time. On these systems, the operating system job scheduler has little or no idea of the requirements of the job. In contrast, JCL explicitly specifies all required datasets and devices.

The scheduler can pre-allocate the resources prior to releasing the job to run. This helps to avoid "deadly embrace", where job A holds resource R1 and requests resource R2, while concurrently running job B holds resource R2 and requests R1. In such cases the only solution is for the computer operator to terminate one of the jobs, which then needs to be restarted. With job control, if job A is scheduled to run job B will not be started until job A completes or releases the required resourcesJ.

Job Processing

05mins [SAQ]

I want you to know that a job is a unit of work which can be made up of many job steps. Each job step is specified in the Job Control Language (JCL) through a set of **Job Control Statements.**

Also know that the Operating System uses **Job Entry System (JES)** to receive jobs into the Operating System, to schedule them for processing, and to control the output.

Job processing goes through a series of steps as given below:

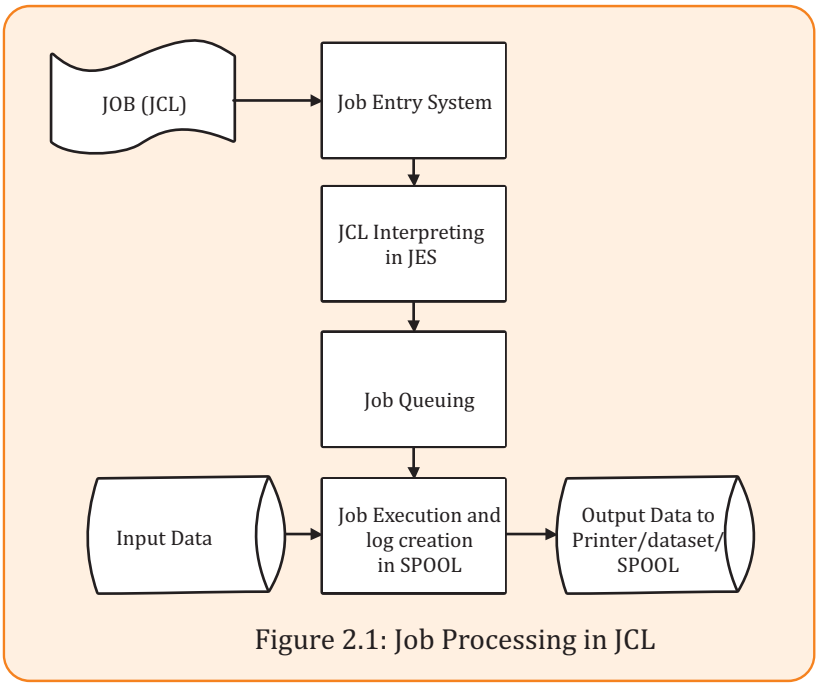


Figure 2.1: Job Processing in JCL

1. **Job Submission** - Submitting the JCL toJES.

2. **Job Conversion** - The JCL along with the PROC is converted into an interpreted text to be understood by JES and stored into a dataset, which we call as SPOOL.

3. **Job Queuing** - JES decides the priority of the job based on CLASS and PRTY parameters in the JOB statement. The JCL errors are checked and the job is scheduled into the job queue if there are noerrors.

4. **Job Execution** - When the job reaches its highest priority, it is taken up for execution from the job queue. The JCL is read from the SPOOL, the

program is executed and the output is redirected to the corresponding output destination as specified in theJCL.

5 Purging - When the job is complete, the allocated resources and the JES SPOOL space is released. In order to store the job log, we need to copy the job log to another dataset before it is released from theSPOOL.

Summary

You have learnt about the :

Job Control Language which is a scripting language used for communication between the application program, operating system and system hardware. We also discussed the steps required for Job processing.

Self-Assessment Questions

1. State three Job Control Statements.
2. What is Job processing?
3. State three functions of a Job Card.

Tutor Marked Assessment

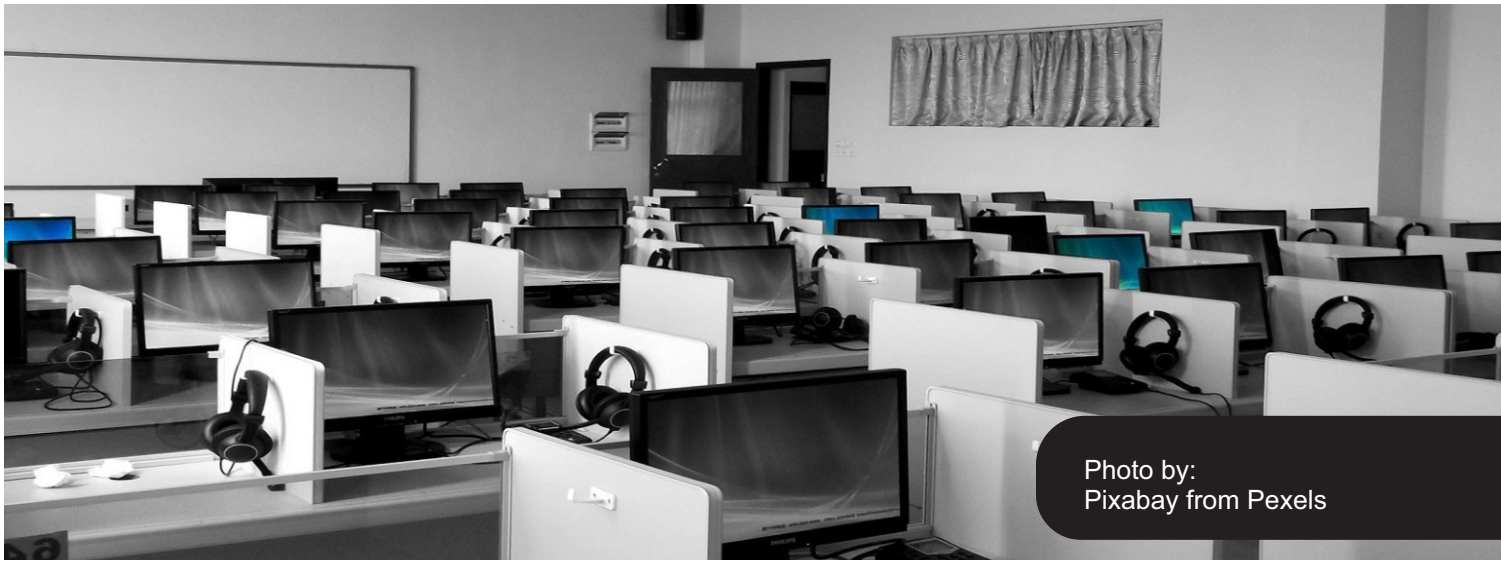
- Explain the steps required for Job processing.
- Explain the reason why Both DOS and OS JCL have a maximum usable line length of 80 characters.

References

Nowosielski. R. J. (n.d): **Job Control Language and the SAS® System for Beginners**, PECO Energy, Philadelphia, PA USA.

Further Reading

- httpswww.tutorialspoint.comjcljcl_tutorial
- https://www.techopedia.com/definition/3378/job-control-language-jcl
- https://en.wikipedia.org/wiki/Job_Control_Language



UNIT 2

Structure of Job Control Language (JCL)

Introduction

The JCL was introduced in the previous unit stating the steps taken for the job processing. In this unit, the structure of Job Control Language is described. You should be very familiar with JCOL so that this unit will be more self-explanatory for you.

Learning Outcomes

At the end of this unit, you should be able to:

- identify program statements used in the JCL, and
- write simple JCL statements.

Main Content

Photo by:
Lorenzo-Herrera
on Unsplash

Structure of JCL

02 mins

I will be presenting the basic structure of a JCL with the common statements below:

//SAMPJCL	JOB1,CLASS=6,MSGCLASS=0,NOTIFY=&SYSUID	(1)
//*		(2)
//STEP010	EXEC PGM=SORT	(3)
//SORTIN	DDDSN=JCL.SAMPLE.INPUT,DISP=SHR	(4)
//SORTOUT	DDDSN=JCL.SAMPLE.OUTPUT,	(5)
//	DISP=(NEW,CATLG,CATLG),DATACLAS=DSIZE50	
//SYSOUT	DDSYSOUT=*	(6)
//SYSUDUM	DD SYSOUT=C	(6)
//SYSPRIN	DDSYSOUT=*	(6)
//SYSIN	DD*	(6)
/*	SORT FIELDS = COPY INCLUDE COND=(28,3, CH, EQ, C “XXX”)	(7)

Program Description

10mins

You should note the numbered JCL statements explained below:

1. **JOB statement-** Specifies the information required for SPOOLing of the job such as job id, priority of execution, user-id to be notified upon completion of the job.

2. **//* statement -** This is a comment statement.

3. **EXEC statement -** Specifies the PROC/Program to be executed. In the above example, a SORT program is being executed (i.e., sorting the input data in a particular order)

4. **Input DD statement -** Specifies the type of input to be passed to the program mentioned in (3). In the above example, a Physical Sequential (PS) file is passed as input in shared mode (DISP =SHR).

5. **Output DD statement -** Specifies the type of output to be produced by the program upon execution. In the above example, a PS file is created. If a statement extendsbeyondthe70thpositioninaline, then it is continued in the next line, which should start with "//" followed by one or more spaces.

6. There can be other types of DD statements to specify additional information to the program (In the above example: The SORT condition is specified in the SYSIN DD statement) and to specify the destination for error/execution log (Example: SYSUDUMP/SYSPRINT). DD statements can be contained in a dataset (mainframe file) or as in-stream data (information hard-coded within the JCL) as given in the above example.

7. /* marks the end of in-stream data.

All the JCL statements except instream data start with //. There should be at least one space before and after JOB, EXEC, and DD keywords and there should not be any spaces in the rest of the statement.

DD STATEMENT

[SAQ1, 2]

It is important to know that the DD (data definition) statement describes a data set and specifies the input and output resources needed for the data set. The SAS system performs similar tasks using the FILENAME and LIBNAME statements.

Also note that the DD statements consist of the characters // in columns 1 and 2, a name field with a maximum of 8 alphanumeric starting in column 3, and the characters DD. A typical format is

```
//MYSAS DD DSN=MY.SAS.DBASE, DISP=SHR
//MYIN DD DSN=MY.INPUT.DATA, DISP= OLD
```

Where:

DSN stands for Data Set Name

DISP stands for disposition.

A file must be allocated before the SAS System can use it. After these statements are processed by the Operating System the data sets are allocated and are available to the SAS system for processing. It means that the Operating System has established a link between the physical data sets - HIGHLEV.MIDLEV. LOWLEV – and the DD Name –MYNAME. When a SAS program needs access to the data set, via an INFILE statement or, a SET Statementin a DATA Step, the data set can be referred to by referring to the DD Name. The following DATA Step, executed in a batch job will read the data stored in the external file called MY.INPUT.DATA and store it in the

```
SAS data set called MY.SAS.DBASE.
Data MYSAS. MYCOPY;
File MYIN;
Input Field1 Field2 Field3;
```

Device independence

From the very beginning, the JCL for the OS family of operating systems offered a high degree of device independence. Even for new files which were to be kept after the end of the job one could specify the device type in generic terms, e.g., UNIT=DISK, UNIT=TAPE, or UNIT=SYSSQ (tape or disk). Of course, if it mattered one could specify a model number or even a specific device address.

Procedures

Procedures permit grouping one or more *"EXEC PGM="* and DD statements and then invoking them with *"EXEC PROC*=procname" -or- simply "EXEC procname" [12]

A facility called a Procedure Library allowed pre-storing procedures.

PROC & PEND

Procedures can also be included in the job stream by terminating the procedure with a // PEND statement, then invoking it by name the same was as if it were in a procedure library.

For example:

```
//SUMPRINT PROC
//PRINTEXEC PGM=IEBGENER
//SYSUT1  DD  DSN=CEO.FILES.DAYEND.RPT24A,DISP=SHR
//SYSUT2  DD  SYSOUT=A
//SYSIN   DD  DUMMY
//PEND
//EXECSUMPRINT
```

Parameterized procedures

You should be aware that OS JCL procedures were parameterized from the start, making them rather like macros or even simple subroutines and thus increasing their reusability in a wide range of situations.

```
//MYPROC PROC FNAME=MYFILE01,SPTYPE=TRK,SPINIT=50,SPEXT=10,LR=100,BLK=1000
.....
//NEWFILE DD DSN=&FNAME,UNIT=DISK,SPACE=(&SPTYPE,&SPINIT,&SPEXT),
//DCB=(LRECL=&LR,BLKSIZE=&BLK),DISP=(NEW,CATLG,DELETE)
....
```

In this example, all the values beginning with ampersands "&" are parameters which will be specified when a job requests that the procedure be used. The PROC statement, in addition to giving the procedure a name, allows the programmer to specify default values for each parameter. So you could use the one procedure in this example to create new files of many different sizes and layouts. For example:

```
//JOB01JOB.....
//STEP01EXECMYPROCFNAME=JOESFILE,SPTYPE=CYL,SPINIT=10,SPEXT=2,LR=100,BLK=2000
or
//JOB02JOB.....
//STEP01EXECMYPROCFNAME=SUESFILE,SPTYPE=TRK,SPINIT=500,SPEXT=100,LR=100,BLK=5000
```

Refer backs

In multi-step jobs, a later step can use a referback instead of specifying in full a file which has already been specified in an earlier step. For example:

```
//MYPROC.....
//MYPR01EXECPGM=.....
//NEWFILE DD DSN=&MYFILE,UNIT=DISK,SPACE=(TRK,50,10),
//DCB=(LRECL=100,BLKSIZE=1000),DISP=(NEW,CATLG,DELETE)
....
//MYPR02EXECPGM=.....
//INPUT01 DD DSN=*.MYPR01.NEWFILE
```

Here, MYPR02 uses the file identified as NEWFILE in step MYPR01 (DSN means "dataset name" and specifies the name of the file).

Moreso, in jobs which contain a mixture of job-specific JCL and procedure calls, a job-specific step can refer back to a file which was fully specified in a procedure, for example:

```
//MYJOBJOB.....
//STEP01EXECMYPROCUsing a procedure
//STEP02EXECPGM=.....Step which is specific to this job
//INPUT01 DD DSN=*.MYPR01.NEWFILE
```

where DSN=*.STEP01.MYPR01.NEWFILE means "use the file identified as NEWFILE in step MYPR01 of the procedure used by step STEP01 of this job". Using the name of the step which called the procedure rather than the name of the procedure allows a programmer to use the same procedure several times in the same job without confusion about which instance of the procedure is used in the referback.

Comments

Do you know that JCL files can be long and complex, and the language is not easy to read? OS JCL allows programmers to include two types of explanatory comment:

- On the same line as a JCL statement. They can be extended by placing a continuation character (conventionally "X") in column 72, followed by "// " in columns 1–3 of the next line.
- Lines which contain only comment, often used to explain major points about the overall structure of the JCL rather than local details. Comment-only lines are also used to divide long, complex JCL files into sections.

```
//MYJOBJOB.....
//*  Lines containing only comments.
//******** Often used to divide JCL listing into sections ********
//STEP01EXECMYPROCComment 2 on same line as statement
//STEP02EXECPGM=.....Comment 3 has been extended and      X
//overflowsinto another line.
//INPUT01 DD DSN=STEP01.MYPR01.NEWFILE
```

Concatenating input files

Let me tell you that OS JCL allows programmers to concatenate ("chain") input files so that they appear to the program as one file, for example

```
//INPUT01 DD DSN=MYFILE01,DISP=SHR
//DDDSN=JOESFILE,DISP=SHR
//DDDSN=SUESFILE,DISP=SHR
```

Note that the 2nd and third statements have no value in the name field, so OS treats them as concatenations. The files must be of the same basic type (almost always sequential), and must have the same record length, however the block length need not be the same.

You should also bear in mind that in early versions of the OS (certainly before OS/360 R21.8) the block length must be in decreasing order, or the user must inspect each instance and append to the named DD statement the maximum block length found, as in, for example,

```
//INPUT01 DD DSN=MYFILE01,DISP=SHR,BLKSIZE=800
//DDDSN=JOESFILE,DISP=SHR (BLKSIZE assumed to be equal to or less than 800)
//DDDSN=SUESFILE,DISP=SHR (BLKSIZE assumed to be equal to or less than 800)
```

In later versions of the OS (certainly after OS/MVS R3.7 with the appropriate "selectable units") the OS itself, during allocation, would inspect each instance in a concatenation and would substitute the maximum block length which was found.

A usual fallback was to simply determine the maximum possible block length on the device, and specify that on the named DD statement, as in, for example,

```
//INPUT01 DD DSN=MYFILE01,DISP=SHR,BLKSIZE=8000
//DDDSN=JOESFILE,DISP=SHR (BLKSIZE assumed to be equal to or less than 8000)
//DDDSN=SUESFILE,DISP=SHR (BLKSIZE assumed to be equal to or less than 8000)
```

The purpose of this fallback was to ensure that the access method would allocate an input buffer set which was large enough to accommodate any and all of the specified datasets.

Conditional processing

OS expects programs to set a return code which specifies how successful the program thought it was. The most common conventional values

- 0 = Normal - all OK
- 4 = Warning - minor errors or problems
- 8 = Error - significant errors or problems
- 12 = Severe error - major errors or problems, the results (e.g. files or reports produced) should not be trusted.
- 16 = Terminal error - very serious problems, do not use the results!

Be informed that OS JCL refers to the return code as COND ("condition code") and can use it to decide whether to run subsequent steps.

However, unlike most modern programming languages, conditional steps in OS JCL are not executed if the specified condition is true—thus giving rise to the mnemonic, "If it's true, pass on through [without running the code]." To complicate matters further, the condition can only be specified after the step to which it refers. For example:

```
//MYJOBJOB.....
//STEP01EXECPGM=PROG01
....
//STEP02EXECPGM=PROG02,COND=(4,GT,STEP01)
....
//STEP03EXECPGM=PROG03,COND=(8,LE)
....
//STEP04EXECPGM=PROG04,COND=(ONLY,STEP01)
....
//STEP05EXECPGM=PROG05,COND=(EVEN,STEP03)
....
```

Job Parameter Types

Don't forget that each of the JCL statements is accompanied by a set of parameters to help the Operating System in completing the program execution. The parameters can be of two types:

1. **Positional Parameters**

(a) Appears at predefined position and order in the statement. Example: AccountinginformationParametercanappearonlyaftertheJOBkeywordand before the programmer name parameter and the Keyword Parameters. If a positional parameter is omitted, it has to be replaced with acomma.

(b) Positional Parameters are present in JOB and EXEC statements. In the above example, PGM is a positional parameter coded after the EXECkeyword.

2. **Keyword Parameters**

(a) Keyword Parameters are coded after the positional parameters, but can appear in any order. Keyword parameters can be omitted, if not required. The generic syntax is KEYWORD= value. Example: MSGCLASS=X, i.e., the job log is redirected to the output SPOOL after the job completion.

(b) In the above example, CLASS, MSGCLASS, and NOTIFY are keyword parameters of JOB statement. There can be keyword parameters in EXEC statement as well.

Summary

The structure of the JCL program has been described with various statements used such as Job statement, EXEC statement, Input DD, Output DD

Self-Assessment Questions

1. What is a DD statement?
2. What symbol is used for a comment statement

Tutor Marked Assignment

Explain the function of the DD and EXEC statements

References

www.tutorialspoint.com

Retrieved on 6th May, 2019 from https://en.wikipedia.org/wiki/Job_Control_Language

Further Reading

Nowosielski. R. J. (n.d) :**Job Control Language and the SAS® System for Beginners**, PECO Energy, Philadelphia, PA USA.

A computer motherboard

Photo by:
Christian Wiediger on Unsplash

UNIT 3

Input/Output system Architecture

Introduction

With the increasing need for data processing, various aspects of computer file system have been studied previously. This unit establishes the relationship between the input/output devices and the software by explaining the different modes of operation.

Learning Outcomes

At the end of this unit, you should be able to:

- state the different modes of input and output,
- list the devices that can be used for both input and output operations on a Computer,
- state the different modes of input and output, and
- list out the interrupt modes we have.

Main Content

picture: A Computer Desktop
Source: Unsplash.com

Input/Output System Architecture

10Mins [SAQ1,2]

Do you know that the computer system's I/O architecture is its interface to the outside world?

This architecture is designed to provide a systematic means of controlling interaction with the outside world and to provide the operating system with the information it needs to manage I/O activity effectively.

a Input devices:

- Optical input devices-Card Reader, Paper Tape Reader, Optical Character Recognition (OCR), Optical Bar code reader (OBR), Digitizer, Optical Mark Reader
- Magnetic Input Devices-Magnetic Stripe Reader, Magnetic Ink Character Recognition (MICR)
- Screen Input Devices-Touch Screen, Light Pen

b Output devices:

- Card Puncher, Paper Tape Puncher
- Monitor (CRT, LCD, LED)

- Printer (Impact, Ink Jet, Laser, Dot Matrix)
- Plotter
- Analog
- Voice

Devices for both input andoutput – disks, network interfaces.

Modes of IO operation

Computers employ the following four basic modes of IO operation:

1. Programmed mode
2. Polling mode
3. Interrupt mode
4. Direct memory access mode.

We shall discuss each of these modes in some detail now.

1. **Programmed Data Mode**

Note that in this mode of communication, execution of an IO instruction ensures that a program shall not advance till it is completed. To that extent, one is assured that IO happens before anything else happens. As depicted in Figure 3.1, in this mode an IO instruction is issued to an IO device and the program executes in "busy-waiting" (idling) mode till the IO is completed. During the busy-wait period, the processor is continually interrogating to check if the device has completed IO. Invariably the data transfer is accomplished through an identified register and a flag in a processor. For example, in Figure 3.1 depicts how an input can happen using the programmed data mode. First, the processor issues an IO request (shown as 1), followed by device putting a data in a register (shown as 2) and finally the flag (which is being interrogated) is set (shown as 3). The device either puts a data in the register (as in case of input) or it picks up data from the register (in case of output). When the IO is accomplished it signals the processor through the flag. During the busy-wait period, the processor is busy checking the flag. However, the processor is idling from the point of view of doing anything useful. This situation is similar to a car engine which is running when the car is not in motion – essentially "idling".

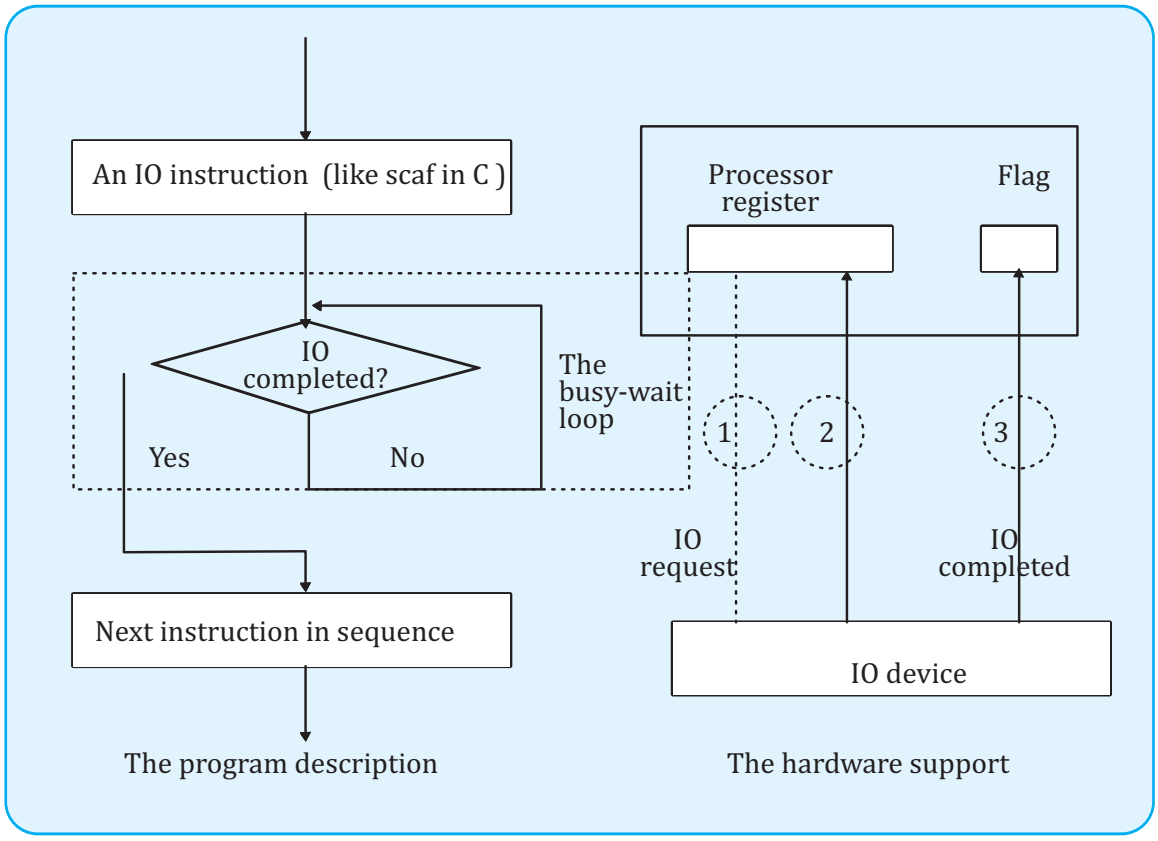


Figure 2.2: Programmed mode of IO.

2 Polling

I want you to know that in this mode of data transfer, the system interrogates each device in turn to determine if it is ready to communicate. If it is ready, communication is initiated and subsequently, the process continues again to interrogate in the same sequence. This is just like a round-robin strategy. Each IO device gets an opportunity to establish Communication in turn. No device has a particular advantage (like say a priority) over other devices.

Polling is quite commonly used by systems to interrogate ports on a network. Polling may also be scheduled to interrogate at some pre-assigned time intervals. It should be remarked here that most daemon software operates in polling mode. Essentially, they use a while true loop as shown in Figure 3.2.

In hardware, this may typically translate to the following protocol:

1. a distinct address to each device connected to a bus.
2. the bus controller scans through the addresses in sequence to find which device wishes to establish a communication.

3. Allow the device that is ready to communicate to leave its data on the register.
4. The IO is accomplished. In case of an input, the processor picks up the data. In case of an output, the device picks up the data.
5. Move to interrogate the next device address in sequence to check if it is ready to communicate.

```
While true
        if device-1 ready to communicate
            then  perform IO with device 1
        if device-2 ready to communicate
            then  perform IO with device 2

        if device-k  ready to communicate
            then  perform IO with device k
end while
```

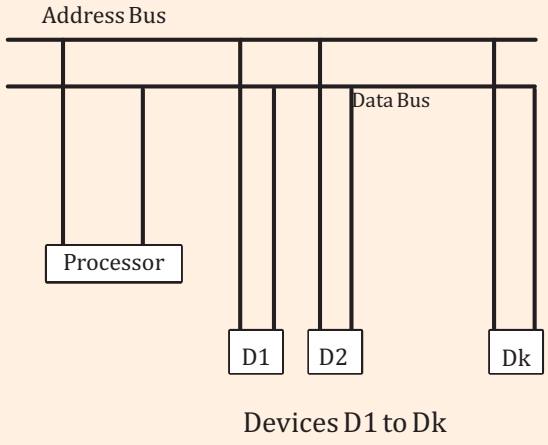


Figure 2.3: Polling mode of IO

Interrupt Mode

[SAQ3,4,5]

Let us begin with a simple illustration to explain the basic rationale behind interrupt mode of data transfer. Suppose a program needs input from a device which communicates using interrupt. Even with the present-day technology the devices are one thousand or more times slower than the processor. If the program waits on the input device it would cycle through many processor cycles just waiting for the input device to be ready to communicate. This is where the interrupt mode of communication scores.

To begin with, a program may initiate IO request and advance without suspending its operation. At the time when the device is ready to establish an IO, the device raises an interrupt to seek communication. Immediately the program execution is suspended temporarily and the current state of the process is stored. The control is passed on to an interrupt service routine (which may be specific to the device) to perform the desired input. Subsequently, the suspended process context is restored to resume the program from the point of its suspension.

Interrupt processing may happen in the following contexts:

a **Internal Interrupt:** The source of interrupt may be a memory resident process or a function from within the processor. We regard such an interrupt as an internal interrupt. A processor malfunction results in an internal interrupt. An attempt to divide by zero or execute an illegal or non-existent instruction code results in an internal interrupt as well. A malfunction arising from a division by zero is called a trap. Internal interrupt may be caused by a timer as well. This may be because either the allocated processor time slice to a process has elapsed or for some reason, the process needs to be pre-empted. Note that an RTOS may pre-empt a running process by using an interrupt to ensure that the stipulated response time required is met. This would also be a case of an internal interrupt.

b **External Interrupt:** If the source of interrupt in not internal, i.e. it is other than a process or processor related event then it is an external interrupt. This may be caused by a device which is seeking the attention of a processor. As indicated earlier, a program may seek an IO and issue an IO command but proceed. After a while, the device from which IO was sought is ready to communicate. In that case, the device may raise an interrupt. This would be a case of an external interrupt.

c **Software Interrupt:** Most OSs offer two modes of operation, the user mode and the system mode. Whenever a user program makes a system call, be it for IO or a special service, the operation must have a transition from user mode to system mode. An interrupt is raised to effect this transition from user to system mode of operation. Such an interrupt is called a software interrupt.

The processor's priority is usually encoded in a few bits of the processor status word. It can be changed by program instructions that write into the program status register (PS). These are privileged instructions, which can be executed only while the processor is running in the supervisor mode.

i The processor is in the supervisor mode only when executing operating system routines. It switches to the user mode before beginning to execute an application program

ii An attempt to execute a privileged instruction while in the user mode leads to a special type of interrupt called a privilege exception

3. Direct Memory Access(DMA) Mode

This is a mode of data transfer in which IO is performed in large data blocks. For instance, the disks communicate in data blocks of sizes like 512 bytes or 1024 bytes. The direct memory access, or DMA ensures access to main memory without processor intervention or support. Such independence from processor makes this mode of transfer extremely efficient.

When a process initiates a direct memory access (DMA) transfer, its execution is briefly suspended (using an interrupt) to set up the DMA control. The DMA control requires the

information on starting address in main memory and size of data for transfer. This information is stored in DMA controller. Following the DMA set up, the program resumes from the point of suspension. The device communicates with main memory stealing memory access cycles in competition with other devices and processor. Figure3.3 shows hardware support.

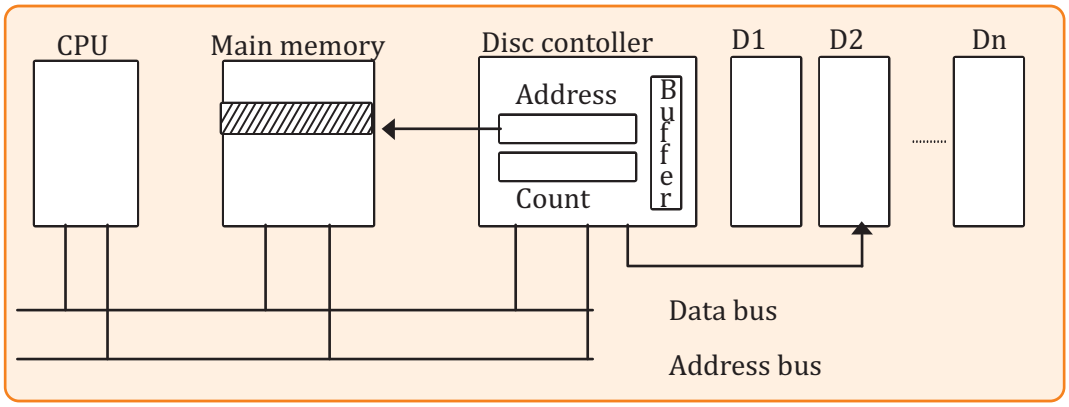


Figure 2.4: DMA (Hardware support)

Let us briefly describe the operations shown in Figure 2.4. Also, we shall assume a case of disk to main memory transfer in DMA mode. We first note that there is a disk controller to regulate communication from one or more disk drives. This controller essentially isolates individual devices from direct communication with the CPU or main memory. The communication is regulated to first happen between the device and the controller, and later between the controller and main memory or CPU if so needed.

Note that these devices communicate in blocks of bits or bytes as a data stream. Clearly, an unbuffered communication is infeasible via the data bus. The bus has its timing control protocol. The bus cannot, and should not, be tied to device transfer bursts. The byte stream block needs to be stored in a buffer isolated from the communication to the processor or main memory.

This is precisely what the buffer in the disk controller accomplishes.

Also note that once the controller buffer has the required data, then one can envisage putting the controller in contention with CPU and main memory or CPU to obtain access to the bus.

Thus, if the controller can get the bus then by using the address and data bus it can directly communicate with the main memory. This transfer shall be completely independent of program control from the processor. So we can effect a transfer of one block of data from the controller to main memory provided the controller has the address where data needs to be transferred and data count of the transfer required. This is the kind of information which initially needs to be set up in the controller address and count registers. Putting this information may be done under a program control as a part of DMA set up. The program that does it is usually the device controller. The device controller can then schedule the operations with much finer control. Data location information in disk

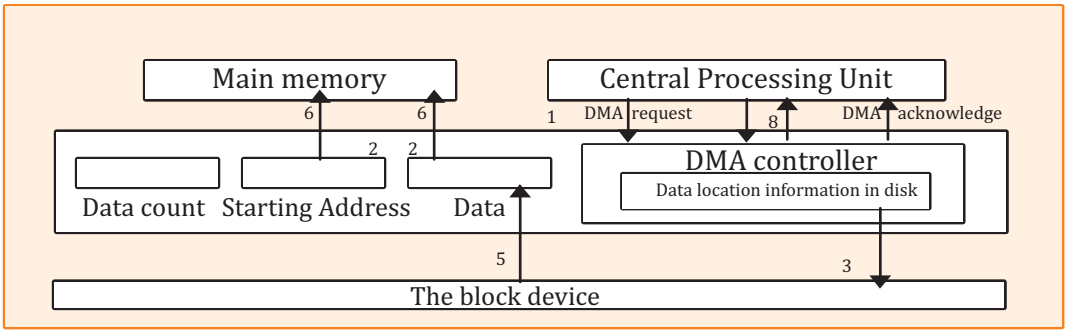


Figure 2.5: DMA: Direct memory access mode of data transfer.

Let us now recap the above mode of transfer within a logical framework with a step-by-step description. This procedure can be understood in the context of Figure 2.5. In the figure, a few lines do not have an accompanying label. However, by following the numbered sequence and its corresponding label, one can find the nature of the operations considered.

The procedure can be understood by following the description given below:

We shall assume that we have a program P which needs to communicate with a device

D and get a chunk of data D to be finally stored starting in address A. The reader should follow through the steps by correlating the numbers in Figure 2.5.

1. The program makes a DMA set-up request.
2. The program deposits the address value A and the data count D, the program also indicates the virtual memory address of the data on disk.
3. The DMA controller records the receipt of relevant information and acknowledges the DMA complete.
4. The device communicates the data to the controller buffer.
5. The controller grabs the address bus and data bus to store the data, one word at a time.
6. The data count is decremented.
7. The above cycle is repeated till the desired data transfer is accomplished. At which time a DMA data transfer complete signal is sent to the process.

The network-oriented traffic (between machines) may be handled in DMA mode. This is so because the network cards are often DMA enabled. Besides, the network traffic usually corresponds to getting information from a disk file at both the ends. Also, because network traffic is in bursts, i.e. there are short intervals of large data transfers. DMA is the most preferred mode of

Typical I/ O Control Steps are stated below

Communication goes across the bus and then:

(a) CPU checks I/O module device status

(b) I/O module returnsstatus

(c) If ready, CPU requests data transfer

(d) I/Omodulegetsdatafromdevice

(e) I/OmoduletransfersdatatoCPU

Summary

In this unit, we have given:

some examples of input and output devices such as mouse, keyboard, scanner, light pen, printer, together with the different operation modes of these devices.

Self-Assessment Questions

1. State the different input/output operation mode you know.
2. Mention a device that can be used for both input and output operation.
3. List out the interrupt modes we have
4. Which is the most preferred mode of operation that supports network traffic
5. An interrupt which is as a result of a memory-resident process is known as?.

Tutor Marked Assessment

- State the difference the Polling mode and the programmed mode
- State four characters that are un acceptable when naming a file
- Explain the usage count attribute of a file.

References

Jin-Fu Li, (n,d),Department of Electrical Engineering National Central University Jungli, Taiwan. Retireved on from http//www.ee.ncu.edu.tw~jflicomputerlecturech05

Further Reading

- https//courses.cs.washington.educoursescse37804spslidesio
- aturing.umcs.maine.edu_meadowcoursescos335COA07

A computer motherboard

Photo by:
Christian Wiediger on Unsplash

UNIT 4

File Naming

Introduction

In this unit, I will be discussing the basic rules of naming a file and its extensions. We will give examples of different types together with their names and extension. Therefore, it is necessary for you to have studied the previous units where Computer file system, types and classifications have been introduced. This will give a good basis for a better understanding of this unit.

Learning Outcomes

At the end of this unit, you should be able to:

- mention some examples of file extensions,
- explain why a file extension is a suffix to the name of a computer file, and
- study the usage of File extension by Operating systems.

Main Content

picture: A Computer Motherboard
Source: Unsplash.com

File Naming and Attributes

(3Mins)

I want you to know that files are abstraction mechanisms. They provide a way to store information and read it back later.

So, a file name is a principal identifier of a file and therefore, good file names provide useful clues to the content, status and version of a file, uniquely identify a file and help in classifying and sorting files. File names which reflect the file content also facilitate searching and discovery of these files.

File Naming Convention

Let me define File Naming Convention (FNC) as a framework for naming your files in a way that describes what they contain and how they relate to other files. Developing an FNC is done through identifying the key elements of the project, the important differences and commonalities between your files. These elements could include things like the date of creation, author's name, project name, name of a section or a sub-section of the project, the version of the file, etc. An advantage to using unique and standardized filenames is the ability to follow path names and link to other systems that

require unique filenames.

You should also note that the exact rules for file naming vary somewhat from system to system, but all operating systems allow strings of one to eight letters as legal file names. The file name is chosen by the person creating it, usually to reflect its contents. There are few constraints on the format of the file name: It can comprise the letters A-Z, numbers 0-9 and special characters $ # & + @!( ) - { } ' ` _ ~ as well as space. The only symbols that cannot be used to identify a file are * | <> \ ^ =? / [ ] ' ; , plus control characters.

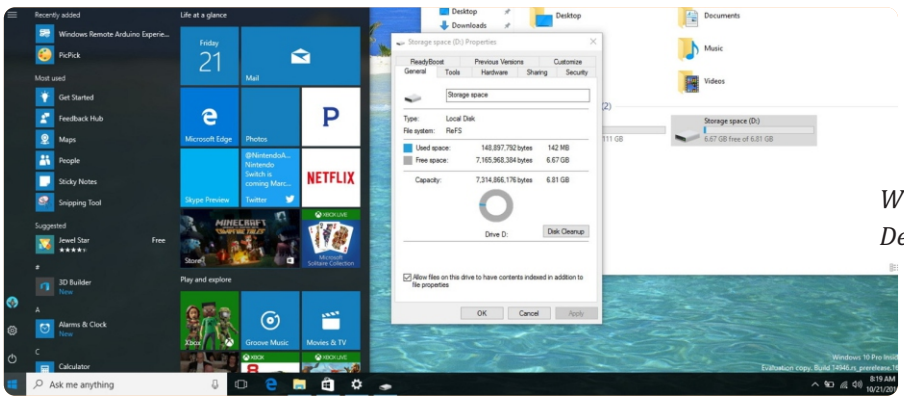
Usually, a file would have two parts with a dot (“.”) separating them. You should note that the part on the left side of the period character is called the **main name** while the part on the right side is called the **extension**. A good example of a file name is “csc604.doc.”

Be informed that file extension differentiates between different types of files. We can have files with same names but different extensions.

It is good to note that when choosing a file name there are different rules for different operating systems that can present problems when files are moved from computer to another. For example,

Microsoft Windows is case insensitive, so files like MYEBOOKS, myebooks, MyEbooks are all the same to Microsoft Windows.

However, under the UNIX operating system, all three would be different files as, in this instance, file names are case sensitive.



Windows 10 Desktop

Elements in a file name

Common elements that you should consider when developing a file naming strategy:

1. Version number (also see 'Data authenticity');
2. Date of creation (date format should be YYYY-MM-DD);
3. Name of creator;
4. Description of content;
5. Name of research team/department associated with the data;
6. Publication date;
7. Project number

Features of a good File Name

- File Names must be Unique
- File Names must Indicate what the file contains
- File Names must Reflect how you work with your information i.ewhat is significant, what is most likely to be searched for, who is the audience.
- File Names must be Naturally ordered
- File Names must be Consistent and understood by everyone

File Name Extension

(3Mins)

You should note that a filename extension is a suffix to the name of a computer file applied to indicate the encoding convention or file format of its contents. In some operating systems (for example UNIX) it is optional, while in some others (such as DOS) it is a requirement. Some operating systems limit the length of the extension (such as DOS and OS/2, to three characters) while others (such as UNIX) do not. Some operating systems (for example RISC OS) do not use file extensions.

Below are several examples of file types and extensions together with the full meaning.

Executable File Extensions you should know

File Extension	File Extension Meaning
.apk	Android Package File
.bat	Batch File
.bin	Binary File
.cgi	Common Gateway Interface Script
.com	MS-DOS command file
.exe	Executable file
.jar	Java Archive File
.py	Python file
.wsf	Windows Script File

Table1.1: Executable file extension and meaning

Audio File Extensions you should know [SAQ1]

File Extension	File Extension Meaning
.aif	AIF/Audio Interchange audio file
.cda	CD audio trach file
.iff	Interchange File Format
.mid or midi	MIDI audio file
.mp3	Mp3 audio file
.mpa	MPEG-2 audio file
.wav	WAVE file
.wma	Windows Media Audio file
.wpl	Windows Media Player playlist

Table 1.2: Audio file extension and meaning

Video File Extensions

File Extension	File Extension Meaning
.avi	Audio Video Interleave File
.flv	Adobe File
.h264	H.264 Video File
.m4v	Apple Mp4 Video FIle
.mkv	Matroska Mutimedia Container
.mov	Apple QuickTime Movie File
.mp4	MPEG-4 Video File
.mpg or mpeg	MPEG Video File
.rm	Real Media File
.swf	Shockwave Flash File
.vob	DVD Video Object File
.wmv	Windows Media Video File

Table 1.3: Video file extension and meaning

Text File Extension

File Extension	File Extension Meaning
.doc and .docx	Microsoft Word File
.odt	OpenOffice Writer Document file
.msg	Outlook Writer Document file
.pdf	Portable Document Format
.rtf	Rich Text Format
.txt	Plain text file
.wpd	WordPerfect document

Table 1.4: Text file extension and meaning

Spreadsheet File Extensions

File Extension	File Extension Meaning
.ods	Ope Office Calc Spreadsheet File
.xlr	Microsoft Works Spreadsheet file
.xls	Microsoft Excel file
.xlsx	Microsoft Excel Open XML Spreadsheet file

Table 1.5: spreadsheet file extension and meaning

Presentation File Extensions

File Extension	File Extension Meaning
.key	Keynote Presentation
.odp	OpenOffice Impress Presentation file
.pps	PowerPoint Slideshow
.ppt	PowerPoint Presentation
.pptx	PowerPoint Open XML Presentation

Table 1.6: Presentation file extension and meaning

Database File Extensions [SAQ2]

File Extension	File Extension Meaning
.accdb	Access 2007 Database File
.csv	Comma Separated Value File
.dat	Data File
.db or .dbf	Database File
.log	Log File
.mdb	Microsoft Access Database File
.pdb	Program Database
.sav	Save File (e.g game save file)

File Extension	File Extension Meaning
.sql	SQL/Structured Query Language database file
.tar	Linux/Unix arball file archive

Table 1.7: Database file extension and meaning

System Related File Extensions

File Extension	File Extension Meaning
.bak	Back-up file
.cab	Windows Cabinet file
.cfg	Configuration file
.cpl	Windows Control Panel file
.cur	Windows Cursor file
.dll	DLL file
.dmp	Dump file
.drv	Device Driver file
.icns	macOS X icon resource file
.ico	Icon file
.ini	Initialization file
.ink	Window Shortcut file
.msi	Windows Installer Package
.sys	Windows System File
.tmp	Temporary file

Table 1.8: system related file extension and meaning

Web File Extensions

File Extension	File Extension Meaning
.asp and aspx	Active Server Page file
.cer	Internal Security Certificate
.cfm	ColdFusion Markup File
.cgi or .pl	Perl Script File
.css	Cascading Style Sheet file
.htm and .html	HTML / Hypertext Markup Language file
.js	JavaScript File
.jsp	Java Server Page file
.part	Partially downloaded file
.php	PHP Source Code file
.rss	RSS / Rich Site Summary file
.xhtml	XHTML / Extensible Hypertext Markup Language file

Table 1.9: Web file extension and meaning

Image File Extensions

File Extension	File Extension Meaning
.ai	Adobe Illustrator
.bmp	Bitmap Image File
.gif	GIF/Graphical Interchange Format
.ico	Icon File
.jpeg or .jpg	JPEG image
.max	3ds Max Scene file
.obj	Wavefront 3D Object File

File Extension	File Extension Meaning
.png	PNG / Portable Network Graphic image
.ps	Partially downloaded file
.psd	PSD/ Adobe Photoshop Document image
.svg	Scalable Vector Graphics file
.tif or .tiff	TIFF Image
..3ds	3D Studio Scene
.3dm	Rhino 3D Model

Table 1.10: image file extension and meaning

Summary

So far in this unit, you have learnt that:

that files provide a way to store information and read it back later, therefore there is a need for a unique Filename which is generally chosen by the person creating it, usually to reflect its contents. It can comprise the letters A-Z, numbers 0-9 and special characters $ # & + @ ! ( ) - { } ' ` _ ~ as well as space. This is followed by a dot and then the file extension. A good example of a file name is "csc604.doc."

File extension differentiates between different types of files and therefore we can have files with same names but different extensions. Different examples of these extensions have been stated.

A computer file is also known to have different attributes like filename, location, type, size, protection, usage count

Self-Assessment Questions

1. State two examples of audio files and give the full meaning of their extension each.
2. State two files indicating their extensions and filenames.
3. Give an example of an Operating system that does not use a File extension

Tutor Marked Assessment

- State four characters that are unacceptable when naming a file
- Explain the usage count attribute of a file.

References

Folder-File-Naming-Convention-10Rules-Best-Practice

Stenzel, N. (n.d): Basic File Management and organization. University of Maryland Extension. Retrieved 17th October 2019 from http//extension.umd.edu

File naming and Organization. Retrieved from https://guides.library.upenn.edu

Further Reading

- lecture10teaching.csse.uwa.edu.auunitsCITS2230handoutsLecture10lecture10.
- https://guides.lib.purdue.edu/c.php?g=353013&p=2378293

A computer motherboard

Photo by:
Christian Wiediger on Unsplash

UNIT 5

File Organization

Introduction

In this unit, you will learn the different File organization methods such as the serial, sequential, indexed-sequential, indexed and direct organization.

Learning Outcomes

At the end of this unit, you should be able to:

- identify which of the forms of file organizations is least complicated,
- identify which file organization has the first field in each record referred to as the key field, and
- state a difference between Pile and sequential files..

Main Content

picture: A Computer Motherboard
Source: Unsplash.com

File Organization

(7Mins)

I want you to know that a file organization deals with the relationship between the control fields of a file record and the physical location of that record in the storage medium.

Types of File Organization

Although a file of records can be arranged in storage media in different ways, all of the ways can be classified by either of two main techniques. They are sequential and random technique.

Sequential

[SAQ1]

Note that sequential order implies that there is a certain order (which can either be numeric or alphanumeric, ascending or descending), of the adjacent records in the file.

Particular fields, located in the same relative positions within the fixed section of all data records of the file, are selected as sort control fields for a specific file sequence.

Random (Non-Sequential)

Also note that a random file organization contains records

stored without regard to the sequence of their record control fields. For example, a random file organization for tape master files is impractical, since the desired record can fall anywhere within the file limits, and each search for a specific record must begin with the first record of the file.

Generally, we have different File Organization methods, these are given below.

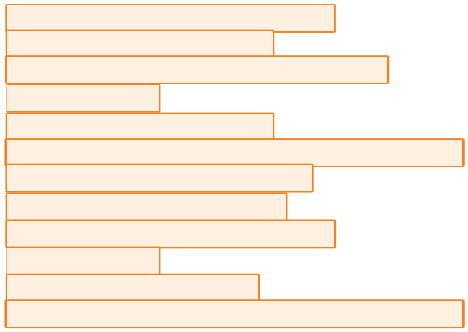
- The pile/serial
- The sequential file
- The indexed sequential file
- The indexed file
- The direct, or hashed file

1 The Pile/Serial [SAQ2]

Be informed that the least-complicated form of file organization may be termed the pile/serial. Data are collected in the order in which they arrive. Each record consists of one burst of data. The purpose of the pile/serial is simply to accumulate the mass of data and save it. Records may have different fields or similar fields in different orders. Thus, each field should be self-describing, including a field name as well as a value. The length of each field must be implicitly indicated by delimiters, explicitly included as a subfield, or known as default for that field type. Because there is no structure to the pile/serial file, record access is by exhaustive search. That is, if we wish to find a record that contains a particular field with a particular value, it is necessary to examine each record in the pile until the desired record is found or the entire file has been searched. If we wish to find all records that contain a particular field or contain that field with a particular value, then the entire file must be searched.

Pile/serial files are encountered when data are collected and stored before processing or when data are not easy to organize. This type of file uses space well when the stored data vary in size and structure; is

perfectly adequate for exhaustive searches, and is easy to update. However, beyond these limited uses, this type of file is unsuitable for most applications.



(a) Pile File

Variable-length records
Variable set of fields
Chronological order

2 The Sequential File

[SAQ2,3]

Do you know that the most common form of file structure is the sequential file? A fixed format is used for records. All records are of the same length, consisting of the same number of fixed-length fields in a particular order. Because the length and position of each field are known, only the values of fields need to be stored; the field name and length for each field are attributes of the file structure. One particular field, usually the first field in each record, is referred to as the **key field.**

The key field uniquely identifies the record; thus key values for different records are always different. Further, the records are stored in key sequence: alphabetical order for a text key, and numerical order for a numerical key.

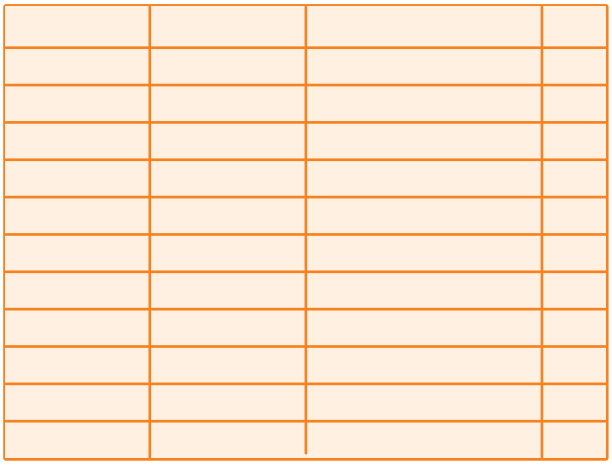
Note that sequential files are typically used in batch applications and are generally optimum for such applications if they involve the processing of all the records. The sequential file organization is the only one that is easily stored on tape as well as disk.

You should also note that for interactive applications that involve queries and/or updates of individual records, the sequential file provides poor performance.

Access requires the sequential search of the file for a key match. If the entire file, or a large portion of the file, can be brought into main memory at one time, more efficient search techniques are possible.

Additions to the file also present problems. Typically, a sequential file is stored in the simple sequential ordering of the records within blocks. That is, the physical organization of the file on tape or disk directly matches the logical organization of the file. In this case, the usual procedure is to place new records in a separate pile file, called a log file or transaction file. Periodically, a batch update is performed that merges the log file with the master file to produce a new file in a correct key sequence.

An alternative is to organize the sequential file physically as a linked list. One or more records are stored in each physical block. Each block on disk contains a pointer to the next block. The insertion of new records involves pointer manipulation but does not require that the new records occupy a particular physical block position. Thus, some added convenience is obtained at the cost of additional processing and overhead.



Fixed - length records
Fixed se of fields in fixed order
Sequential order based on key field

(a) Sequential File

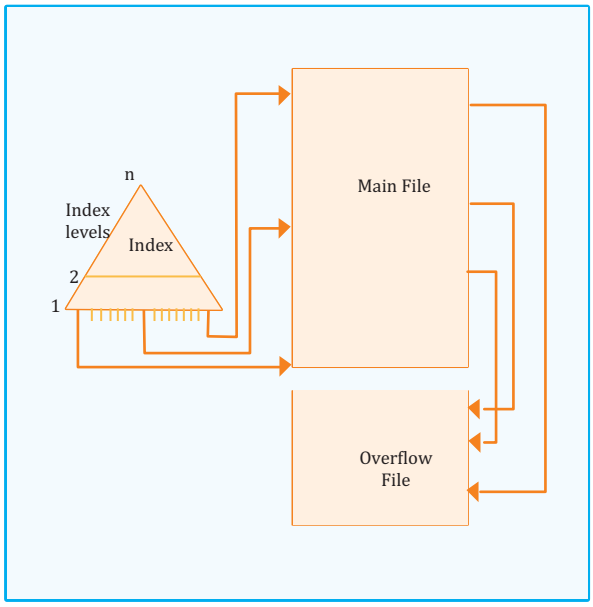
3 The Indexed Sequential File [SAQ2,3]

Be informed that in this type of file system, a popular approach to overcoming the disadvantages of the sequential file is the indexed sequential file. The indexed sequential file maintains the key characteristic of the sequential file: records are organized in sequence based on a key field. Two features are added: an index to the file to support random access, and an **overflow file.**

The index provides a lookup capability to quickly reach the vicinity of a desired record. The overflow file is similar to the log file used with a sequential file but is integrated so that a record in the overflow file is located by following a pointer from its predecessor record.

You should be aware that in the simplest indexed sequential structure, a single level of indexing is used. The index, in this case, is a simple sequential file. Each record in the index file consists of two fields: a key field, which is the same as the key field in the main file, and a pointer into the main file. To find a specific record, the index is searched to find the highest key value that is equal to or precedes the desired key value. The search continues in the main file at the location indicated by the pointer.

Also have it in mind that additions to the file are handled in the following manner: Each record in the main file contains an additional field not visible to the application, which is a pointer to the overflow file. When a new record is to be inserted into the file, it is added to the overflow file. The record in the main file that immediately precedes the new record in logical sequence is updated to contain a pointer to the new record in the overflow file. If the immediately preceding record is itself in the overflow file, then the pointer in that record is updated. As with the sequential file, the indexed sequential file is occasionally merged with the overflow file in batch mode.



The indexed sequential file greatly reduces the time required to access a single record, without sacrificing the sequential nature of the file. To process the entire file sequentially, the records of the main file are processed in sequence until a pointer to the overflow file is found, then accessing continues in the overflow file until a null pointer is encountered, at

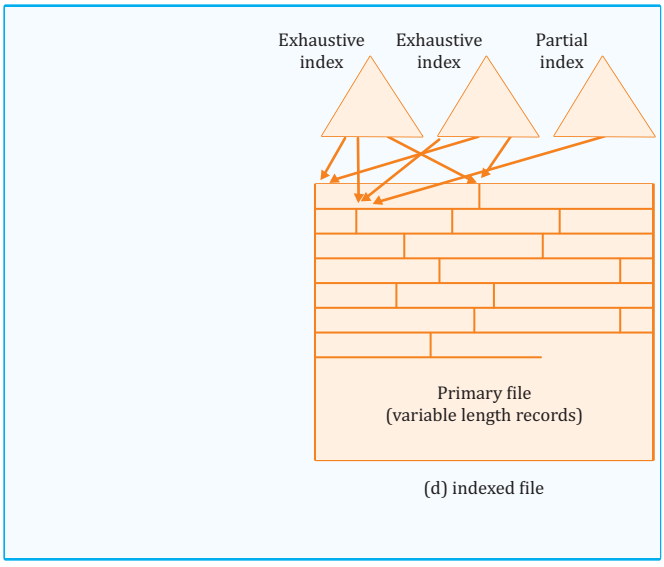
To provide even greater efficiency in access, multiple levels of indexing can be used. Thus the lowest level of the index file is treated as a sequential file and a higher-level index file is created for that file. which time accessing of the main file is resumed where it left off.

4 The Indexed File

Put at the back of your mind that the indexed sequential file retains one limitation of the sequential file: effective processing is limited to that which is based on a single field of the file. For example, when it is necessary to search for a record based on some other attributes than the key field, both forms of the sequential file are inadequate. In some applications, the flexibility of efficiently searching by various attributes is desirable.

For us to achieve this flexibility, a structure is needed that employs multiple indexes, one for each type of field that may be the subject of a search. In the general indexed file, the concept of sequentiality and a single key are abandoned. Records are accessed only through their indexes. The result is that there is now no restriction on the placement of records as long as a pointer in at least one index refers to that record. Furthermore, variable-length records can be employed.

Two types of indexes are used. An exhaustive index contains one entry for every record in the main file. The index itself is organized as a sequential file for ease of searching. A partial index contains entries to records where the field of interest exists. With variable-length records, some records will not contain all fields. When a new record is added to the main file, all of the index files must be updated. Indexed files are used mostly in applications where timeliness of information is critical and where data are rarely processed exhaustively. Examples are airline reservation systems and inventory control systems.



(d) indexed file

5 The Direct or Hashed File

Let me say that the direct or hashed file exploits the capability found on disks to access directly any block of a known address. As with sequential and indexed sequential files, a key field is required in each record. However, there is no concept of sequential ordering here. The direct file makes use of hashing on the key value. Direct files are often used where very rapid access is required, where fixed length records are used, and where records are always accessed one at a time. Examples are directories, pricing tables, schedules, and name lists.

Summary

So far in this unit, you have learnt that:

File organization deals with the relationship of the control fields of a file record to the physical location of that record in the storage medium. The different File Organization methods that have been explained are: Pile, Sequential, Indexed sequential, Indexed and Direct.

Self-Assessment Questions

1. Explain the Sequential File Organization method
2. State a difference between Pile and sequential files
3. What is a Key Field?

Tutor Marked Assessment

- Give two (2) examples of sequential file organization method

References

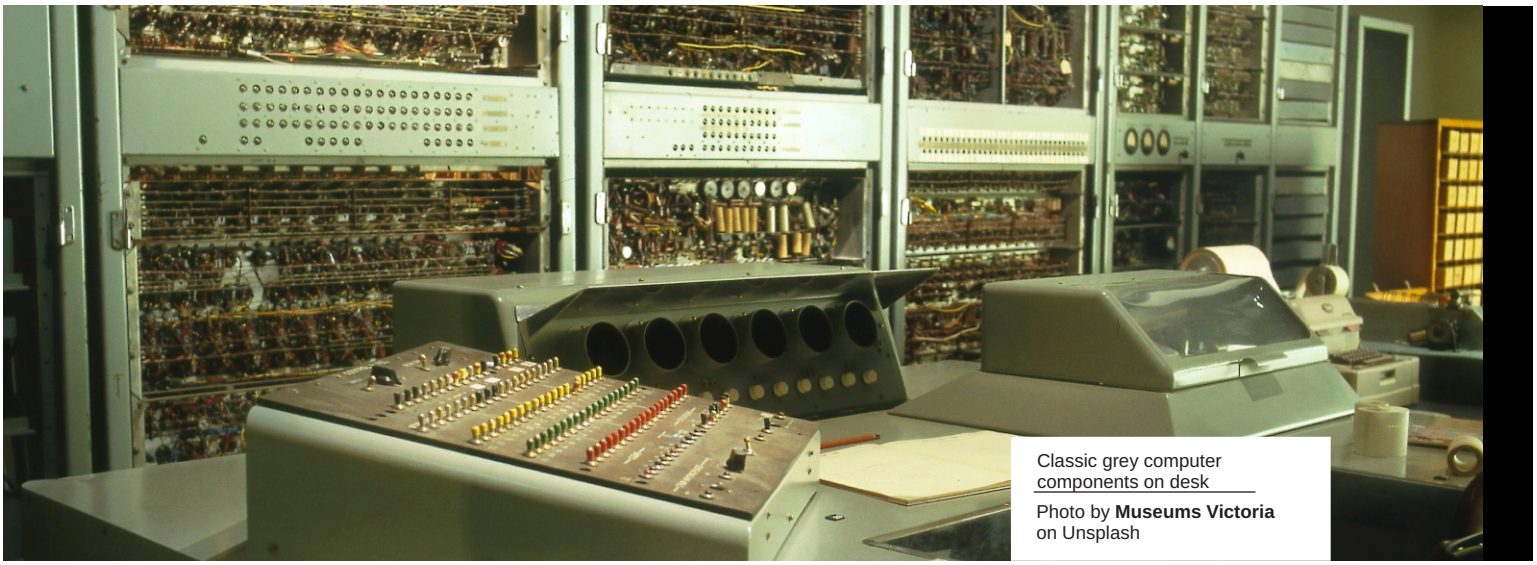
- Stenzel, N. (n.d): Basic File Management and organization. University of Maryland Extension. Retrieved 17th October 2019 from http//extension.umd.edu
- File naming and Organization. Retrieved from https://guides.library.upenn.edu

Further Reading

- Data Processing Technique Handbook for IBM, Retrieved from http://bitsavers.trailing-edge.com › C20-1638-1_Data_File_Handbook_Mar66

Module 3

Mapping logical organization onto Physical Storage & Backup procedure and file recovery



Classic grey computer components on desk

Photo by **Museums Victoria** on Unsplash

UNIT 1

Mapping logical organization onto Physical Storage

Introduction

You now have a general knowledge about the Computer file system; its operations, its communications with the hardware, the different ways they are organized, its naming procedure from the previous module/unit. You should now be able to understand the lesson on File Directories for this unit. This unit, therefore, introduces the mapping of logical file organizations into physical storage.

Learning Outcomes

At the end of this unit, you should be able to:

- state different directory structures, and
- mention five operations of file system.

Main Content

File Directories

07mins [SAQ1]

Do you know that associated with any file management system and collection of files is a file directory?

The directory contains information about the files, including attributes, location, and ownership. Much of this information, especially those that concern storage, is managed by the operating system. The directory is itself a file, accessible by various file management routines. Although some of the information in directories is available to users and applications, this is generally provided indirectly by system routines.

Concept of File Directory

Also have it in mind that to keep track of files, the file system normally provides directories, which, in many systems are themselves files. The structure of the directories and the relationship among them are the main areas where file systems tend to differ, and it is also the area that has the most significant effect on the user interface provided by the file system

Contents of File Directory

Now, from the user's point of view, the directory provides a mapping between file names, known to users and applications, and the files themselves. Thus, each file entry includes the name of the file. Virtually all systems deal with

different types of files and different file organizations, and this information is also provided.

Also note that an important category of information about each file concerns its storage, including its location and size. In shared systems, it is also important to provide information that is used to control access to the file.

Typically, one user is the owner of the file and may grant certain access privileges to other users. Finally, usage information is needed to manage the current use of the file and to record the history of its usage.

Logical Structure of Directory [SAQ2]

The directories can be structured in the following ways:-

1. Single level directory
2. Two level directory
3. Tree structured directory
4. Acyclic graph directory
5. General graph directory

1 **Single level directory:** In a single level directory system, all the files are placed in one directory. This is very common on single-user OSs. A single-level directory has significant limitations, however, when the number of files increases or when there is more than one user. Since all files are in the same directory, they must have unique names. If there are two users who call their data file "test", then the unique-name rule is violated. Although file names are generally selected to reflect the content of the file, they are often quite limited in length. Even with a single-user, as the number of files increases, it becomes difficult to remember the names of all the files to create only files with unique names shown

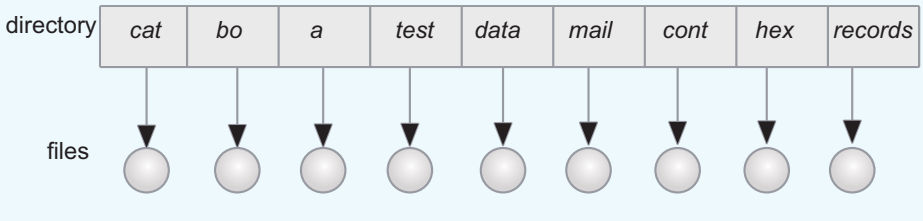


Figure 3.1 Single level Directory

2 **Two level directory:** In the two-level directory system, the system maintains a master block that has one entry for each user. This master block contains the addresses of the directory of the users. There are still problems with two level directory structure. This structure effectively isolates one user from another. This is an advantage when the users are completely independent, but a disadvantage when the users want to cooperate on some task and access files of other users. Some systems simply do not allow local files to be accessed by other users shown

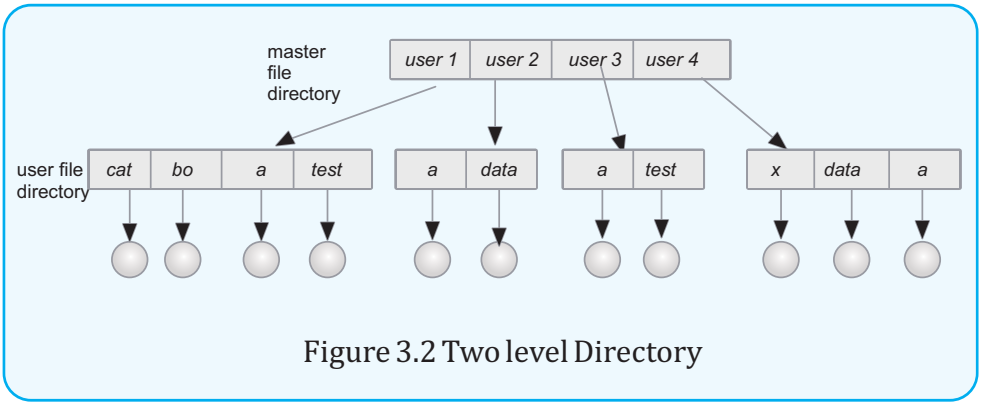


Figure 3.2 Two level Directory

3 **Tree structured directory:** In the tree-structured directory, the directory themselves are files. This leads to the possibility of having sub-directories that can contain files and sub-sub directories. An interesting policy decision in a tree-structured directory structure is how to handle the deletion of a directory. If a directory is empty, its entry in its containing directory can simply be deleted. However, suppose the directory to be deleted id not empty, but contains several files, or possibly sub-directories. Some systems will not delete a directory unless it is empty. Thus, to delete a directory, someone must first delete all the files in that directory. If these are any sub-directories, this procedure must be applied recursively to them, so that they can be deleted also. This approach may result in an insubstantial amount of work shown below

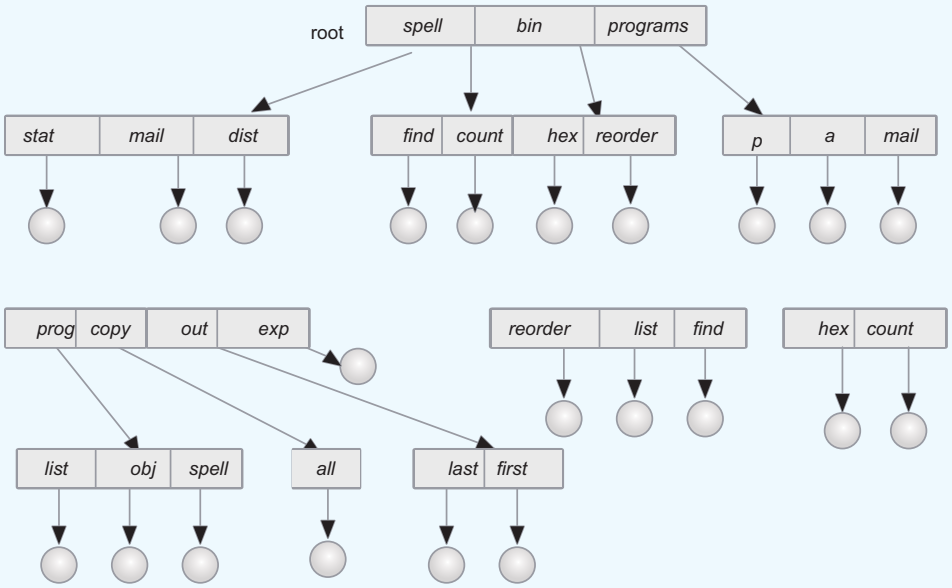


Figure 3.3 Tree structured Directory

4

Acyclic graph directory: The acyclic directory structure is an extension of the tree-structured directory structure. In the tree-structured directory, files and directories starting from some fixed directory are owned by one particular user. In the acyclic structure, this prohibition is taken out and thus a directory or file under directory can be owned by several users shown

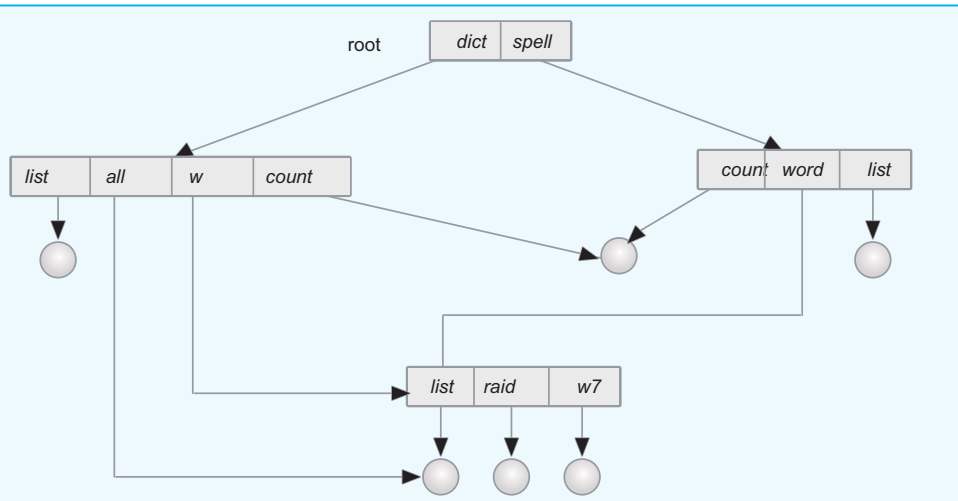


Figure 3.4 Acyclic graph Directory

5 **General graph directory:** The general graph directory is formed by adding links into an existing tree structure. It overcomes the problem of acyclic graph by allowing the cycles in a directory. Thus it avoids the searching of a component twice in a subdirectory.

Path Names [SAQ3]

Do you know that when a file system is organized as a directory tree, some way is needed for specifying the filenames? The use of a tree-structured directory minimizes the difficulty in assigning unique names. Any file in the system can be located by following a path from the root or master directory down various branches until the file is reached. The series of directory names, culminating in the file name itself, constitutes a **pathname** for the file. Two different methods commonly used are:

- Absolute Path name
- Relative Path name

Absolute Path Name

Be informed that each file is given a path which consist of the path from the root directory to the file. Let us take this as an example, the file in the lower left hand corner of Figure below has the pathname User_B/Word/Unit_A/ABC.

The slash is used to delimit names in the sequence. The name of the master directory is implicit, because all paths start at that directory. Note that it is perfectly acceptable to have several files with the same file name, as long as they have unique pathnames, which is equivalent to saying that the same file name may be used in different directories. In this example, there is another file in the system with the file name ABC, but that has the pathname /User_B/Draw/ABC.

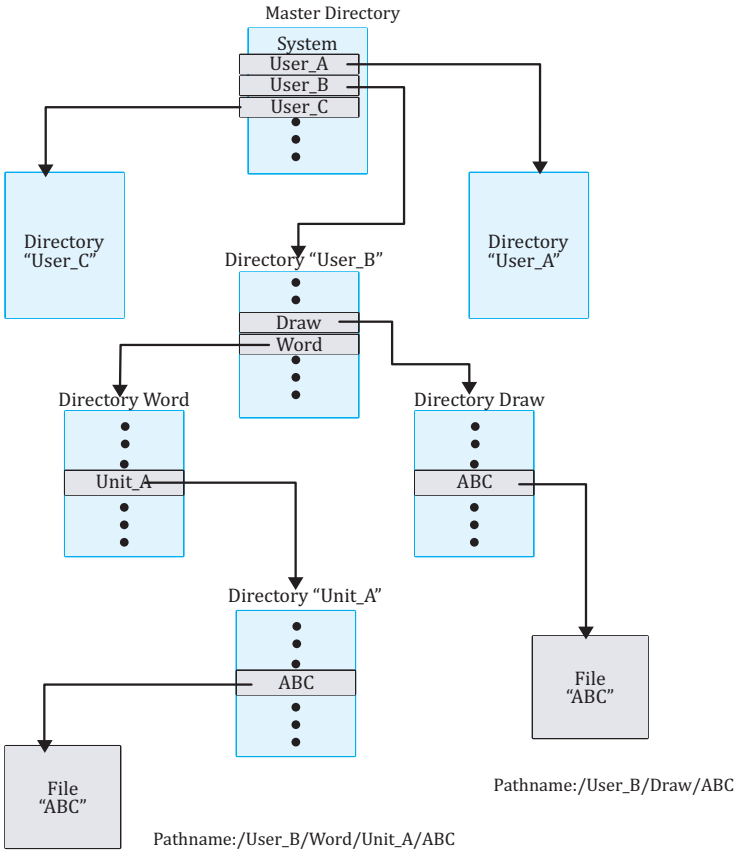


Figure 3.5 Absolute path name Directory

Note that absolute file names always start at the root directory and are unique.

Relative Path Name

Although the pathname facilitates the selection of file names, it would be awkward for a user to have to spell out the entire pathname every time a reference is made to a file. Typically, an interactive user or a process has associated with it a current directory often referred to as the working directory or current directory.

Files are then referenced relative to the working directory. For example, if the working directory for user B is "Word," then the pathname Unit_A/ABC is sufficient to identify the file in the lower left-hand corner of Figure above. When an interactive user logs on, or when a process is created, the default for the working directory is the user home directory. During execution, the user can navigate up or down in the tree to change to a different working directory

File Protection

You should be aware that files often contain information that is highly valuable to their users. One of the major functions of the file system is to protect this information against unauthorized access and physical damage. Physical damage may occur because of of hardware problems, power failure, head crashes, dirt and extreme temperatures. To prevent such damage some systems perform backup at regular intervals.

Protection is achieved by limiting the type of file access which can be made. Access is permitted or denied depending upon several factors, one of which is the type of access requested. Several operations on files can be controlled. Some of these are:

a. **read** - read a file

b. **write** - write a file

c. **execute** - load and execute a file

d. **append** - append information at the end of a file

e. **delete** - free the space allocated to a file

f. **update** – modifying, deleting and adding to a file

g. **copy**- copy the contents of a file

h. **list**- listing the name of the file

I want you to know that the most common implementation of the file systems allows the owners of the file to do operations 1-5, whereas other users can only invoke those operations that do not modify the file, e.g., file read. However, in some systems,e.g., UNIX, the user can change the access control of a file such that he can let anybody access (modification allowed)the file or he can completely deny any user (including himself) access to a file. Files use different access rights such as:

1 **Access Control:**-Let me tell you that access control is the most common approach to protect the files and directories depending upon the identity of the users. Access control limits who can access your files and how they can access them. Users and group of users are granted certain access rights to a file. An access list is associated with each file or directory. The access list contains information on the type of users and accesses that they can do on a directory or file. An example is the following access list associated with a

UNIX file or directory:

Drwxrwxrwx

The **D** indicates that this is an access list for a directory, the first **rwx** indicates that it can be read, written, and executed by the owner of the file, the second **rwx** is an access information for users belonging to the same group as the owner (somewhere on the system is a list of users belonging to the same group as the owner) and the last **rwx** for all other users. The **rwx** can be changed to just r--indicating that it can only be read, or -w- for write-only, --xfor execute only.

2 **Password Protection**:-Another approach to protect your file from unauthorized access is to use a password with each file. This scheme associates a password to each file. If you does not know the password associated with a file then you cannot access it. This is a very effective way of protecting files but for a user who owns many files, and constantly changes the password to make sure that nobody accesses these files will require that users have photographic memories.

3 **File Naming**:-This depends upon the inability of a user to access a file he cannot name. This can be implemented by allowing only users to see the files they have created.

Be aware that since most file systems allow only a limited number of characters for filenames, there is no guarantee that two users will not use the same filenames. A name is attached to every file to uniquely identify it and access it through its name. The exact rules for naming file vary from the system but all the operating systems allow a string of one to eight letters as legal filename

Summary

In this unit, we have explained:

the file directory, with the logical structure listed and explained. You also learnt several other operations that can be performed by a file as we as their access rights. We also consider how your files can be protected without anyone having access to it.

Self-Assessment Questions

1. Give any three File access right you know,
2. Briefly explain the two level directory, and
3. Mention two Path names.

Tutor Marked Assessment

1.Explain the File operations you know

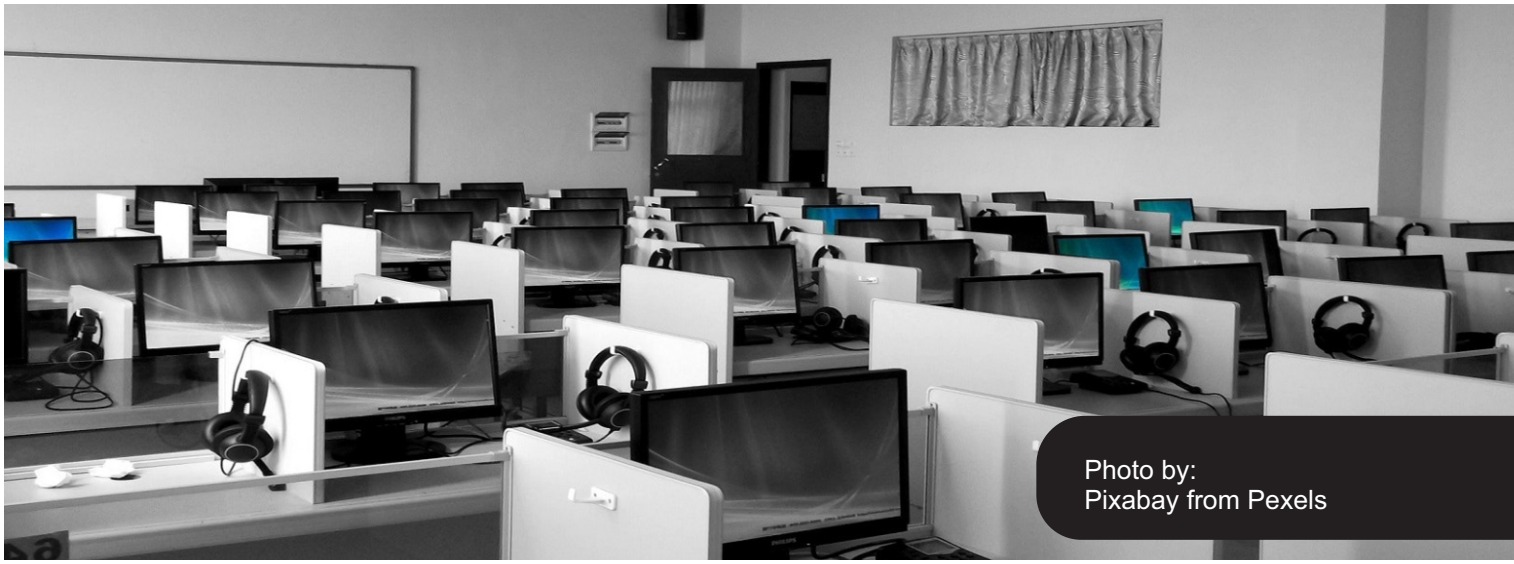
2.With the aid of diagram, explain the **TWO-LEVEL** File Directory

References

M. Kaur, S. Singh, & R. Kaur, (2016): Directory Structure and File Allocation

Further Reading

Allocation Methods.International Journal of Computer Science and Information Technologies (IJCSIT), Vol. 7(2), pp577-582



UNIT 2

File Allocation Methods

Introduction

You have learnt File directory structures in the previous unit. In this unit, I will introduce File allocation methods such as contiguous, linked and indexed allocation.

Learning Outcomes

At the end of this unit, you should be able to:

- state the different major methods of allocating disk space to files
- state the advantages and disadvantages of each of the allocation method.
- discuss what is meant by File allocation method

.

Main Content

Photo by:
Lorenzo-Herrera
on Unsplash

File Allocation Methods

07mins [SAQ1]

Let me tell you what allocation means, Allocation refers to the process of assigning secondary storage space in files. The files should be allocated space in such a manner so that disk space is utilized effectively and files can be accessed quickly. The allocation method is responsible for mapping a file's logical blocks into the actual physical blocks on the secondary storage device. In most operating systems, the size of a physical block is a power of 2 between 512 and 4096. We have three major methods of allocating disk space to files, namely:

1. Contiguous allocation
2. Linked allocation
3. Indexed allocation

1 **Contiguous allocation**:-The contiguous allocation method requires each file to occupy a set of contiguous address on the disk. Disk addresses define a linear ordering on the disk. Notice that, with this ordering, accessing block b+1 after block b normally requires no head movement. When head movement is needed (from the last sector of one cylinder to the first sector of the next cylinder), it is only one track. Thus, the number of disk seeks required for accessing contiguous allocated files in minimal, as is seek time when a seek is finally needed.

Contiguous allocation of a file is defined by the disk address and the length of the first block. If the file is nblocks long and starts at location b, then it occupies blocksb, b+1, b+2, ..., b+n-1. The directory entry for each file indicates the address of the starting block and the length of the area allocated for this file. The difficulty with a contiguous allocation is finding space for a

new file. If the file to be created is n blocks long, then the OS must search for n free contiguous blocks. First-fit, best-fit, and worst-fit strategies are the most common strategies used to select a free hole from the set of available holes. Simulations have shown that both first-fit and best-fit are better than worst-fit in terms of both time storage utilization. Neither first-fit nor best-fit is best in terms of storage utilization, but first-fit is generally faster. These algorithms also suffer from external fragmentation. As files are allocated and deleted, the free disk space is broken into little pieces.

External fragmentation exists when enough total disk space exists to satisfy a request, but this space not contiguous; storage is fragmented into a large number of small holes.

The operating system that uses contiguous allocation is IBM VM/CMS. For example file count starts from 0 and length is 2, so the end is 1, file tr starts from 14 and length is 3 so the end is 16, file mail starts from 19 and ends 24, the file list starts from 28 and ends with 31 and file f starts from 6and end with 8.

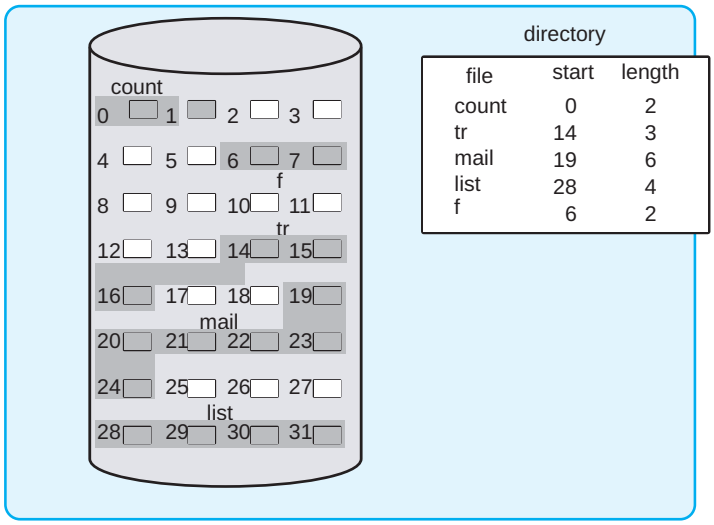


Figure 3.6 Contiguous Allocation

Advantages of Contiguous Allocation SAQ 2

1 It is simple to implement because keeping track of where a file's blocks are reduced to remembering only one number.

2 Performance is good because the entire file can be read from the disk in a single operation.

3 In this scheme, the number of disk seeks required for accessing the file is minimal. Disk addresses define a linear ordering on the disk, accessing block

b+1 after block b requires no head movement. As a result, the number of disk seeks required for a file is less.

4 It is the best from the point of view of the individual sequential file.

Disadvantages of Contiguous Allocation

1 Contiguous allocation method suffers internal as well as external fragmentation as files are allocated and deleted, the free disk space is broken into little pieces.

2 In terms of memory utilization, this method is inefficient.

It is difficult to increase the file size because it depends on the availability of contiguous memory

2 **Linked allocation**:-The problems we have in contiguous allocation can be traced directly to the requirement that the spaces be allocated contiguously and that the files that need these spaces are of different sizes. These requirements can be avoided by using linked allocation. In linked allocation, each file is a linked list of disk blocks. The directory contains a pointer to the first (and optionally the last) block of the file. For example, lets say we have a file of 5 blocks which starts at block 9, might continue at block 16, then block 1, block 10 and finally block 25. Each block contains a pointer to the next block and the last block contains a NIL pointer. The value -1 may be used for NIL to differentiate it from block 0. With linked allocation, each directory entry has a pointer to the first disk block of the file. This pointer is initialized to nil (the end-of-list pointer value) to signify an empty file.

A write to a file removes the first free block and writes to that block. This new block is then linked to the end of the file. To read a file, the pointers are just followed from block to block. There is no external fragmentation with linked allocation. Any free block can be used to satisfy a request. Notice also that there is no need to declare the size of a file when that file is created. A file can

continue to grow as long as there are free blocks. Linked allocation, does have disadvantages, however. The major problem is that it is inefficient to support direct-access; it is effective only for sequential-access files. To find the ith block of a file, it must start at the beginning of that file and follow the pointers until the ith block is reached.

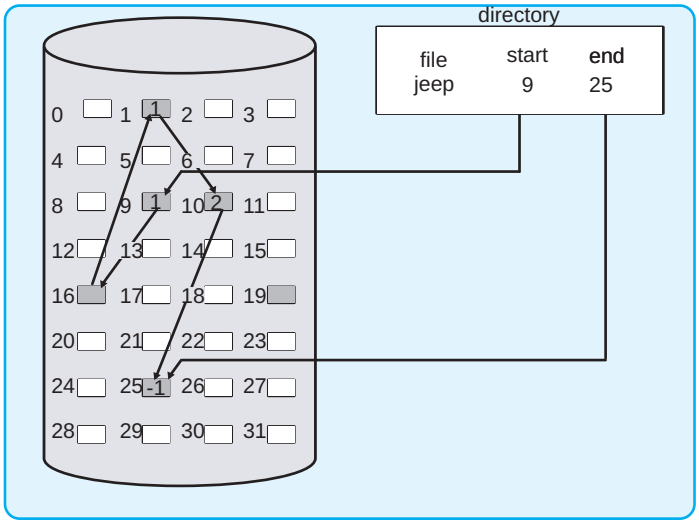


Figure 3.7 Linked Allocation

Advantages of Linked Allocation:

1. Unlike contiguous allocation every free disk block can be utilized.
2. It does not suffer from the problem of external fragmentation.
3. There is no need to declare the size of a file at the time of its creation. A file can grow as long as free blocks are available.
4. There is no need to perform the compaction.

Disadvantages of Linked Allocation

1. In this scheme, there is large no of seeks because the file blocks are randomly distributed on disk.
2. Linked allocation is comparatively slower than contiguous allocation.
3. Random or direct access is not supported by this scheme we cannot access the blocks directly.
4. The pointer is extra overhead on the system due to the linked list.

3 **Indexed Allocation**: - Note that the indexed allocation method is the solution to the problem of both contiguous and linked allocation. This is done by bringing all the pointers together into one location called the index block. Of course, the index block will occupy some space and thus could be considered as an overhead of the method. In indexed allocation, each file has its index block, which is an array of disk sector of addresses. The ith entry in the index block points to the ith sector of the file.

The directory contains the address of the index block of a file. To read the ith sector of the file, the pointer in the ith index block entry is read to find the desired sector. Indexed allocation supports direct access, without suffering from external fragmentation. Any free block anywhere on the disk may satisfy a request for more space. For example, set the index value is 19 all blocks are linked to that index values provide the references to all blocks. Index method follows the direct method using the index block.

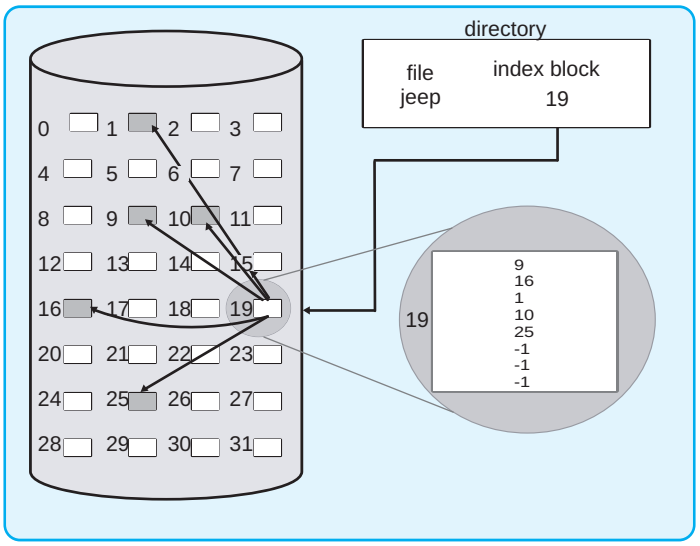


Figure 3.8 Indexed Allocation

Advantages of Indexed Allocation:

1. It is the most popular form of file allocation and support both sequential and direct access to the file.
2. Any free block on the disk can be used for allocation.
3. Allocation of space and the basis of individual block eliminates external fragmentation.
4. Allocation of space based on variable size portions improves locality.

Disadvantages of Linked Allocation

SAQ3

1. If the index block is small, it will not be able to hold enough pointers for a large file.
2. The entire index or table will have to be kept in main memory for all the times to make it work.
3. Looking up for an entry in a large index is a time consuming process.
4. Indexed allocation suffers from the wasted space.
5. For the large size file, it is very difficult for a single index block to hold all the pointers.

For very large files, single index block may not be able to hold all the pointers. Following mechanisms can be used to resolve this:

1. **Linked scheme:** This scheme links two or more index blocks together for holding the pointers. Every index block would then contain a pointer or the address to the next index block.

2. **Multilevel index:** Try to understand that in this policy, a first level index block is used to point to the second level index blocks which in turn points to the disk blocks occupied by the file. This can be extended to 3 or more levels depending on the maximum file size.

3. **Combined Scheme:** Also be aware that in this scheme, a special block called the **Inode (information Node)** contains all the information about the file such as the name, size, authority, etc and the remaining space of Inode is used to store the Disk Block addresses which contain the actual file as shown in the image below. The first few of these pointers in Inode point to the **direct blocks** i.e the pointers contain the addresses of the disk blocks that contain data of the file. The next few pointers point to indirect blocks. Indirect blocks may be single indirect, double indirect or triple indirect. Single Indirect block is the disk block that does not contain the file data but the disk address of the blocks that contain the file data. Similarly, **double indirect blocks** do not contain the file data but the disk address of the blocks that contain the address of the blocks containing the file data.

Summary

We have studied File allocation methods in this unit. These are the Contiguous, Indexed and Linked allocation methods and we have also discussed the advantages and the disadvantages of all the allocation methods.

Self-Assessment Questions

1. Mention the three major methods of allocating disk space to files
2. State one advantage and disadvantage of Contiguous allocation

Tutor Marked Assignment

State the difference between the contiguous allocation and the indexed allocation.

With the aid of a diagram show how the Contiguous File allocation works.

References

Retrieved on 11th October 2020 from https://www.includehelp.com/operating-systems/file-allocation-method.aspx.

Allocation Methods. International Journal of Computer Science and Information Technologies (IJCSIT), Vol. 7(2), pp577-582

Further Reading

M. Kaur, S. Singh, & R. Kaur, (2016): Directory Structure and File

A computer motherboard

Photo by:
Christian Wiediger on Unsplash

UNIT 3

Back up and Disaster Recovery

Introduction

Computer File system requires a lot of data processing which has been discussed in the previous units. File protection and storage mechanisms have also been discussed. However, there is a limit to these protections such that disasters like fire outbreak, flood and some other self-caused disasters can get these files destroyed permanently. Therefore, there is a need for a file backup and also a recovery plan. In this unit, we will extensively discuss these.

Learning Outcomes

At the end of this unit, you should be able to:

- outline two types of disasters that can affect a computer file system,
- state another name for business continuity plan, and
- note the types of data replications.

Main Content

picture: A Computer Desktop
Source: Unsplash.com

Backup and Disaster Recovery

03Mins [SAQ1,2]

Generally, there are two kinds of computer users. Those that have lost a major chunk of data, and those who are going to lose a major chunk of data.

Data loss can be due to natural disasters, theft, human error, and computer failure. The most common loss of data among users is due to the "loss" of data somewhere on the computer. The best way to prevent such loss is to know the physical location of your data (local drive, removable media, network) and to use logical file names.

Therefore, a **backup** is a copy of data from your database that can be used to construct that data. Data has changed. There is more of it, it's larger and is no longer allocated on a single storage device in the middle of the data center. As a result, backup is now reaching the breaking point. Just one example of how this is manifesting itself is that some application owners and users no longer trust their data to IT, enterprises are using outsourcing for backup their data.

You should be aware that backups can be divided into two:

1. **Physical backups** are backups of the physical files used in storing and recovering your database, such as data files, control files, and archived redo logs. Ultimately, every physical backup is a copy of files storing database

information to some other location, whether on disk or some offline storage such as tape.

2. **Logical backups:** contain logical data exported from a database with an Oracle export utility and stored in a binary file, for later re-importing into a database using the corresponding Oracle import utility.

Therefore, it is essential for us to back-up all data (e.g., data files, code books, software settings, computer programs, word processing documents). Backup systems entail manual or automated copying of files to removable media(e.g., floppy disks, Zip disks, tape) or to network storage. Backup procedures should be thoroughly tested to ensure archived files remain uncorrupted and can be restored. Procedures should be written up so that personnel unfamiliar with backup and restore methods could follow them from scratch.

However, we have cases of disasters which can be classified into two broad categories. The first is **natural disasters** such as floods, hurricanes, tornadoes or earthquakes. While preventing a natural disaster is impossible, risk management measures such as avoiding disaster-prone situations and good planning can help. The second category is **man-made disasters**, such as hazardous material spills, infrastructure failure, bio-terrorism, and disastrous IT bugs or failed change implementations. In these instances, surveillance, testing and mitigation planning are invaluable.

Components involved in backup

10mins

a. **Backup Controller:** (cvs) It's the main object in the life of one backup operation. It contains all the information needed to be able to execute a backup successfully. From the complete specs of the backup all the settings, logging, output options and also the whole backup plan to be executed and various steps that any backup must fulfil.

b. **Backup Loggers:** (cvs) Typical loggers chainable eto log information to various places. Each backup controller has one chain of loggers instantiated.

c. **Backup Destinations** (cvs) (not implemented yet): The basic idea about backup destinations is to be able to send the final backup file to/over a variety of systems.

d. **BackupSettings:** (cvs) Anything altering backup behaviour must be

considered to be one set. No matter if it has visual representation in the UI or not or if it can be configured by the user or not.

e **Backup Structure:** (cvs) The base component on top of which all the rest of backup is built. Implements one simple PHP API allowing to define (virtually) any structure suitable to be sent to xml, fetch any information from the iterator and perform other operations transparently and consistently. In three words, "the heart of backup".

f **Backup Plan:** (cvs and cvs2) The execution plan, dependent on the type and Format of the backup. Split into tasks, each one having one or more steps. There are two types of steps: execution steps that only execute (custom) phpcode, used for anything or being XML output and structure steps that, using the "backup structure" component, are used basically to generate XML output.

May I interest you to know that disaster recovery is primarily a form of Long Distance state replication combined with the ability to start up applications at the backup site after a failure is detected. Incomplete RTOs and RPOs can quickly derail a disaster recovery plan. Every item in the DR plan requires a defined recovery point and time objective, as failure to create them may lead to significant problems that can extend the disaster's impact. Once the RTO and RPO metrics have been mapped to IT infrastructure, the DR planner can determine the most suitable recovery strategy for each system. The organization ultimately sets the IT budget and therefore the RTO and RPO metrics need to fit with the available budget. While most business unit heads would like zero data loss and zero time loss, the cost associated with that level of protection may make the desired high availability solutions impractical. A cost-benefit analysis often dictates which disaster recovery measures are implemented.

The amount and type of state that is sent to the backup site can vary depending on the application's needs. State replication can be done at one of these layers

1 within an application,

2 per disk or within a file system, or

3 for the full system context.

Replication at the application layer can be the most optimized, only transferring the crucial state of a specific application. Backup mechanisms operating at the file system or disk layer replicate all or a portion of the file system tree to the remote site without requiring specific application knowledge. The use of virtualization makes it possible to not only transparently replicate the complete disk, but also the memory context of a virtual machine allowing it to seamlessly resume operation after a failure. However, such a technique is typically designed only for LAN environments due to significant bandwidth and latency requirements.

In general DR services fall under one of the following categories: Hot Backup Site provides a set of mirrored standby services that are always available to run the application once a disaster occurs, providing minimal Recovery Time Objective (RTO) and Recovery Point Objective (RPO). Hot Stand-by Site typically uses synchronous replications to prevent any data loss due to disaster. This form of backup is the most expensive since fully powered servers must be available at all times to run the application, plus extra licensing fees may apply for some applications. Warm Backup Sites may keep the state up to date with either synchronous or asynchronous replication schemes depending on the necessary RPO. Stand-by servers to run the application after failure are available, but are kept in a warm state where it may take minutes to bring them online. This slows recovery but also reduces cost. The server resources to run the application must be available at all times. Cold Back-up Site data is often only replicated periodically, leading to an RPO of hours and days.

Besides, servers to run the application after failure are not readily available and there may be a delay of hours or days as hardware is kept in storage until needed. In addition to managing state replication, a DR solution must be able to detect when a disaster has occurred, perform a failure procedure to activate the backup site, as well as run the fallback steps necessary to revert the control to the primary data centre once the disaster has been dealt with.

Recovery Point Objective

Recovery Point Objective (RPO) of a DR system represents the point of time of the most recent backup prior to any failure. For some applications, absolutely no data can be lost (RPO=0) requiring continuous synchronous replications to be used. While for other applications the acceptable data loss could range from a few seconds to hours or even days.

Recovery Time Objective

Recovery Time Objective (RTO) of a DR is an orthogonal business decision that specifies a limit on how long it can take for an application to come back online after a failure occurs. This includes the time to detect the failure, prepare any required servers in the backup site (virtual or physical), initialize the failed application, and perform the network reconfiguration required to reroute requests from the original site to the backup site so that the application can be used. Depending on the application type and backup technique, this may involve additional manual steps such as verifying the integrity of the state or performing application-specific data restore operations, and can require careful scheduling of recovery tasks to be done efficiently.

Performance

For a DR service to be useful it must have a minimal impact on the performance of each application being protected under failure-free operation. DR can impact performance either directly such as in a synchronous replication case where an application write will not return until it is committed remotely, or indirectly by simply consuming disk and network bandwidth resources which otherwise the application could use.

Consistency

The DR service must ensure that after a failure occurs the application can be restored to a consistent state. This may require the DR mechanism to be application-specific to ensure that all relevant state is properly replicated to the backup site. In other cases, the DR system may assume that the application will keep a consistent copy of its important state on disk and use a disk replication scheme to create consistent copies at the backup site. It is important you note that the primary and backup sites are geographically separated to ensure that a single disaster will not impact both sites. This geographic separation adds its challenges since increased distance leads to higher WAN bandwidth costs and will incur greater network latency. Also bear in mind that increased roundtrip latency directly impacts application response time when using synchronous replications. Asynchronous techniques can improve performance over longer distances but can lead to greater data loss during a disaster.

Disaster recovery planning [SAQ3]

Disaster recovery planning is a subset of a larger process known as business continuity planning and includes planning for resumption of applications,

3. Direct Memory Access(DMA) Mode

data, hardware, electronic communications (such as networking) and other IT infrastructure. A business continuity plan (BCP) includes planning for non-IT related aspects such as key personnel, facilities, crisis communication and reputation protection, and should refer to the disaster recovery plan (DRP) for IT related infrastructure recovery/continuity.IT disaster recovery control measures can be classified into the following three types:

(1) Preventive measures - Controls aimed at preventing an event from occurring.

(2) Detective measures - Controls aimed at detecting or discovering unwanted events.

(3) Corrective measures - Controls aimed at correcting or restoring the system after a disaster or an event.

You should bear in mind that a business continuity action plan is a document that contains and controls critical information that a business needs to stay running in spite of adverse events. A business continuity plan is also called an emergency plan. A good business continuity plan should clearly state the business's essential functions in writing. Note that an information technology disaster recovery plan (IT DRP) should be developed in conjunction with the business continuity plan. Priorities and recovery time objectives for information technology should be developed during the business impact analysis. Technology recovery strategies should be developed to restore hardware, applications and data in time to meet the needs of the business recovery.

The document should identify and prioritize which systems and processes must be sustained and provide the necessary information for maintaining them. A business continuity action plan should include the following information: Employer Contact List, Key Supplier/Vendor Information, Key Contacts, Prioritized List of Critical Business Functions, Recovery Locations,

Copies of Essential Records, Critical Telephone Numbers, Critical Supplies List, Inventory of the Company's Equipment/Machinery/Vehicles, Inventory of the Company's Computer Equipment and Software, List of Communication Venues, Data Response Plan.

Be informed **Data replication(DR)** is the process of copying data from one location to another. Replication helps an organization pass up to date copies of its data in the event of a disaster. Replication can take place at the host, in the array, or over the network. Replication can take place over a storage area network, local area network, or local wide area network, as well as in a cloud. Cloud computing platforms are well suited for offering DR as a service due to their use of automated virtual platforms that can minimize the recovery time after a failure. For disaster recovery (DR) purposes, replication typically occurs between a primary storage location and a secondary offsite location. Host based replication uses servers to copy data from one site to another and is designed to allow a virtual machine to continue to function in times of disaster. With the array-based application, compatible storage arrays use built in software to automatically copy data between arrays. Network based data replication requires a switch or appliance between storage arrays and servers. We have two types of data replications.

Synchronous replication takes place in real-time, and asynchronous replication is time delayed. Synchronous replication is preferred for applications with low recovery time objectives that cannot lose data, but it is more expensive and creates latency that slows down the primary application. Asynchronous replication is designed to work over distances and requires less bandwidth. Because there is a delay in the copy time, the two copies of data may not always be identical with asynchronous replication. Replication is often combined with snapshot technology which allows users to replicate data periodically while still being able to roll back to a specific point in time for recovery. Deduplication, which eliminates redundant data is also frequently combined with replication for DR and backup.

Let me bring to your notice that Server virtualization is a driver for disaster recovery because virtualization reduces the number of servers required for a disaster recovery site. Virtual servers are stored as files or virtual machine (VM) images on the host and can be moved by copying the VM image file and booting it on another host while physical servers require the same hardware at the DR site. Tools for replicating virtual machines include PHD Virtual esXpress, VizioncorevReplicator or VMware Site Recovery Manager if your array supports it, or tools built into applications such as Oracle that replicate data between servers.

The cloud also fits with replication, because it can remove cost and complexity from disaster recovery. It alleviates the need to acquire and manage an off-site location. Host-based replication is generally the best fit for disaster recovery through the cloud because storage array and network based replication require devices at the source and target locations. Host based replication lets you move data from standard servers in your environment to the provider's servers off-site.

May I let you know that a critical aspect of disaster recovery planning is often overlooked. Disaster recovery testing. Disaster recovery testing is the only reliable way for an organization to gauge the effectiveness of its disaster preparedness, and data recovery planning. Simply verifying that a backup can be restored is not enough.

Do you know that we have significant differences between data restoration and business continuity?

Disaster recovery testing covers a range of services. It must demonstrate the ability to recover data, as well as quicklyreturn of applications infrastructure components and mission critical systems to an operational state following a disaster. IT pros must work to develop an effective DR test plan while establishing criteria for evaluating the metrics that are gathered during recovery testing. DR testing also allows you to conduct planned maintenance, offers a training opportunity for staff, and creates awareness within an organization about disaster recovery procedures.

Prior to selecting a disaster recovery strategy, a disaster recovery planner first refers to their organization's business continuity plan which should indicate the key metrics of recovery point objective (RPO) and recovery time objective (RTO) for various business processes (such as the process to run

payroll, generate an order, etc.). The metrics specified for the business processes are then mapped to the underlying IT systems and infrastructure that support those processes.

Also note that recovery strategies should be developed for Information technology (IT) systems, applications and data. This includes networks, servers, desktops, laptops, wireless devices, data and connectivity. Priorities for IT recovery should be consistent with the priorities for recovery of business functions and processes that were developed during the business impact analysis. IT resources required to support time-sensitive business functions and processes should also be identified.

The recovery time for an IT resource should match the recovery time objective for the business function or process that depends on the IT resource. Information technology systems require hardware, software, data and connectivity. Without one component of the "system," the system may not run. Therefore, recovery strategies should be developed to anticipate the loss of one or more of the following system components:

a **Computer room environment** (secure computer room with climate control, conditioned and backup power supply, etc.).

b **Hardware** (networks, servers, desktop and laptop computers, wireless devices and peripherals).

c **Connectivity to a service provider** (fibre, cable, wireless, etc.).

d **Software applications** (electronic data interchange, electronic mail, enterprise resource management, office productivity, etc.).

Summary

You have learnt in this unit:

that all too often computer users save files to unknown locations (usually the default set up by the program) but never find saved files or have the saved files deleted by the local area network as a part of routine data cleanup. Therefore, it is essential to back-up all data and procedures

should be thoroughly tested to ensure that archived files remain uncorrupted and can be restored. You also learnt that disaster recovery control measures can be classified into Preventive measures, Detective measures and Corrective measures. And we also discussed data replication which is the process of copying data from one location to another

Self-Assessment Questions

1. Mention two types of data replications
2. Explain one of the mentioned types above
3. What is Disaster Recovery Testing?

Tutor Marked Assessment

Explain the function of the following in DR systems

a.Recovery time Objective

b.Recovery point Objective

References

Retrieved on 12th October 2019, from http://encyclopedia.thefreedictionary.com/Intrusion-prevention-system

Retrieved on 12th October 2019, from https://www.ready.gov/business/implementation/IT Official website of the Department of Homeland Security

Further Reading

http://searchstorage.techtarget.com/definition/Business-Continuity-and-Disaster-Recovery

https://en.wikipedia.org/wiki/Disaster_recovery#Further_reading

A computer motherboard

Photo by:
Christian Wiediger on Unsplash

UNIT 4

Recovery Strategies

Introduction

In the previous unit, we discussed backup strategies for file systems. Generally, when there is a file backup there must also be a recovery plan. We will therefore introduce some file recovery strategies in this unit

Learning Outcomes

At the end of this unit, you should be able to:

- list recovery strategies for small sized business applications, and
- describe the recovery strategy used by larger business

Main Content

picture: A Computer Desktop
Source: Unsplash.com

Recovery Strategies

12Mins [SAQ1]

Tips

May I interest you to know that some business applications cannot tolerate any downtime. They utilize dual data centres capable of handling all data processing needs, which run in parallel with data mirrored or synchronized between the two centres.

This is a very expensive solution that only larger companies can afford. However, there are other solutions available for small to medium sized businesses with critical business applications and data to protect. They are :

Internal Recovery Strategies.

Many businesses have access to more than one facility. Hardware at an alternate facility can be configured to run similar hardware and software applications when needed. Assuming data is backed up off-site or data is mirrored between the two sites, data can be restored at the alternate site and processing can continue.

Vendor Supported Recovery Strategies.

You should be aware that some vendors can provide “hot sites” for IT disaster recovery. These sites are fully configured data centres with commonly used hardware and software products. Subscribers may provide unique equipment or software either at the time of disaster or store it at the

time of disaster or store it at the hot site ready for use. Data streams, data security services and applications can be hosted and managed by vendors. This information can be accessed at the primary business site or any alternate site using a web browser. If an outage is detected at the client site by the vendor, the vendor automatically holds data until the client's system is restored.

These vendors can also provide data filtering and detection of malware threats, which enhance cyber security. Data streams, data security services and applications can be hosted and managed by vendors. This information can be accessed at the primary business site or any alternate site using a web browser. If an outage is detected at the client site by the vendor, the vendor automatically holds data until the client's system is restored. You should also be aware thar these vendors can also provide data filtering and detection of malware threats, which enhance cyber security.

Note that a backup and restore strategy contains a backup portion and a restore portion. The backup part of the strategy defines the types and frequencies of backups, the nature and speed of the hardware that is required for them, how backups are to be tested, and where and how backup media is supposed to be stored (including security considerations). The restore part of the strategy defines who is responsible for performing restores and how restores should be performed to meet the goals for the availability of the database and for minimizing loss. Backup and restore operations occur within the context of a recovery model.

A **recovery model** is a database property that controls how the transaction log is managed. The recovery model also determines what types of backups and what restore scenarios are supported for the database. Typically, a database uses either the simple recovery model or the full recovery model. The full recovery model can be supplemented by switching to the bulk-logged recovery model before a bulk operation.

Test the backup and recovery procedures thoroughly before a real failure

occurs. Testing helps ensure that you have the required backup to recover from various failures, that the procedures are clearly defined and documented, and can be executed smoothly and quickly by any qualified operator. Perform regular database and transaction log backups to minimize the amount of lost data. Backup both system and user databases. Securely maintain system logs. Keep records of all service packs installed in Microsoft and SQL server. Keep records of network libraries used and the security mode. A documented copy of the backup and restore procedures should be kept in a run book.

In addition to preparing for the need to recover systems, organizations also implement precautionary measures to prevent a disaster in the first place. These may include:

1. Local mirrors of systems and/or data and use of disk protection technology such as RAID.
2. Surge protectors to minimize the effect of power surges on delicate electronic equipment.
3. Use of an uninterruptible power supply (UPS) and/or backup generator to keep systems going in the event of a power failure.
4. Fire prevention/mitigation systems such as alarms and fire extinguishers, and
5. Anti-virus software and other security measures. Recent research supports the idea that implementing a more holistic pre-disaster planning approach is more cost-effective in the long run. Every $1 spent on hazard mitigation (such as a disaster recovery plan) saves society $4 in response and recovery costs.

As IT systems have become increasingly critical to the smooth operation of a company, and arguably the economy as a whole, the importance of ensuring the continued operation of those systems, and their rapid recovery, has increased. For example, of companies that had a major loss of business data, 43% never reopen and 29% close within two years. As a result, preparation

for continuation or recovery of systems needs to be taken very seriously. This involves a significant investment of time and money to ensure minimal losses in the event of a disruptive event.

It should not be forgotten that data backup and recovery are not the same. For one thing, the backup software can fail, or the person responsible for backing up the data can fail. Backing up data without recovery in mind is tantamount to not backing up the data at all. There are other steps that have to be taken to successfully restore the data in an event where it is needed. Steps like assembling the right recovery environment, (the right operating system, servers, and storage), and the right people, procedures and tools to bring back the backed up data. Backup software can fail. Data has to be backed up as if it will be needed one day. From a backup perspective, the main concern is not restoration, it is to back up data as quickly as possible. Getting a secure copy of the data backed up at an offsite location is only the first step of disaster recovery.

A second step requires having the right recovery system connected to the data, which means a need for the right servers, storage, hypervisors, and operating system in the recovery environment. Basically, the recovery environment needs to reflect the production environment. This is not an easy step, as many changes occur daily in the production environment that IT staffs are frequently too busy to capture. The last step is having the right people, processes, and tools needed to recover at the time when they are needed. All of this is to say that data back up and disaster recovery is not the same, but both are necessary for long term business technology resiliency. Having a recovery mindset is a necessity, which means backing up data according to recovery strategy, connecting the right recovery systems to the properly backed up data, and creating a programmatic approach to recovery by positioning with the right people, right processes, right tools, and making sure that they are all available at the right time.

Intrusion Prevention Systems

In response to the changing threat landscape, Network Intrusion Prevention Systems were developed to provide advanced protection beyond that offered by firewalls and Intrusion Detection Systems (IDS). Firewalls and IDS provides security but do not provide the kind of protection that an IPS provides. IPS is a technology that provides security for computer systems with features that are effective in facing threats in their advance stage. IPS can detect attacks whether they are known or unknown. IPS is also a network security device that monitors the network and/or system activities for unwanted behaviour and can interact to prevent these activities. IPS is considered an important component in ant IT system defence. IPS protects from denial of service attacks (DOS) and prevents intrusions that target software applications. It is designed to operate completely invisible on a network. IPS products do not typically claim an IP address but can respond directly to any traffic in a variety of ways.

Intrusion Prevention Systems are considered as extensions of Intrusion Detection Systems because both monitor network traffic and/or system activities for malicious activity. The main difference between the two systems is unlike IDS, IPSs are placed in-line and can actively prevent and block intrusions that are detected. More specifically IPS can take such actions as sending an alarm, dropping malicious packets, resetting the connection and/or blocking traffic from the offending IP address. IPS can also correct Cyclic Redundancy Check errors, unfragment packet streams, prevent TCP sequencing issues, and clean up unwanted transport and network layer options. Let me bring to your notice that IPS can be classified into four different types.

1. Network based intrusion prevention systems (NIPS), monitors the entire network for suspicious traffic by analyzing protocol activity.

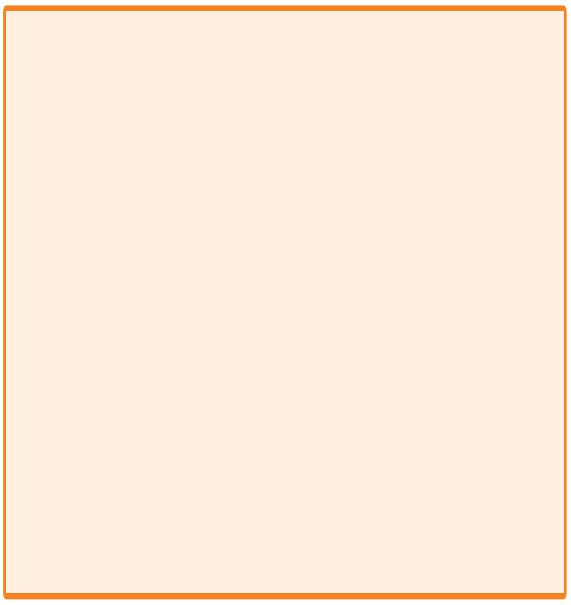
2. Wireless intrusion prevention systems (WIPS) monitors a wireless network for suspicious traffic by analyzing wireless network protocols.

3. Network behavioural analysis (NBA) examines network traffic to identify threats that generate unusual traffic flows, such as distributed denial of service (DDos) attacks, certain forms of malware and policy violations. Host based intrusion prevention system (HIPS) is an installed software package which monitors a single host for suspicious activity, by analyzing events occurring within that host. The majority of intrusion prevention systems

4 utilizes one of three detection methods.

Signature-Based Detection monitors packets in the network and compares the packets with pre-configured and pre-determined attack patterns known as signatures. Statistical Anomaly Based Detection determines the normal network activity, like what sort of bandwidth is generally used, what protocols are used, what ports and devices generally connect, and alerts the administrator or user when traffic is detected that is anomaly (not usual).

Stateful Protocol Analysis Detection is the method which identifies deviations of protocol states by comparing observed events with predetermined profiles of generally accepted definitions of benign activity. IPS is a very effective technique to protect databases and networks from unauthorized users. Like other developments, it has its limitations, but the limitations are heavily outweighed by the advantages. Combining network and host IPS techniques to protect databases and networks creates a robust defensive prevention. Combining IPS IDS, and firewall technologies will provide a strong defence line which can protect systems from any and every attack. It is an outstanding protection and prevention scheme for any computer.



For the medical record company disaster recovery plan, I would have a generator in place in case of power outages, I would use external hard drives and written logs as on-site back-up with written procedures to follow, and off-site cloud service as a major backup database in case of a severe disaster. Backup recovery initial information would include records of employer contact list, key supplier/vendor information, key contacts, prioritized list of critical business functions, recovery locations, copies of essential records, critical telephone numbers, critical supplies list, inventory of the company's equipment/machinery/vehicles, inventory of the company's computer equipment and software, list of communication venues, and the technical aspects of the recovery procedure. This would be done at all four locations with the major backup system located at an off-site cloud

storage. All four offices would have an in-house IT technician who would be responsible for backing up the database system regularly and who would also be comfortable with the disaster recovery procedure. There would also be two other company employees with the knowledge of what procedures needs to be followed to obtain a successful database recovery at all times in case of an absence. This disaster recovery plan would be implemented in all four offices.

My decision on choosing external hard drives as choice backup media is because, in the event of a small natural disaster, the hardware, software, and personnel will all be readily available. This is all the office needs. A cloud platform will be in place in case of a major disaster, and just as not needing a generator the same size as a hospital, an elaborate backup system would be too cost effective and maybe even time effective as going through complicated recovery procedures would just make the recovery time more difficult to achieve.

Each office would have a database with updated backup information and recovery procedures for all four of the offices. This would not be a hard task to accomplish when there is an employee hired specifically for the task. I would use a firewall system, an IDS system, and an IPS system combined for protection from intrusion threats bought on by hackers, cyber-threats and from employee related mistakes. This is an area that has to be given priority because the database has personal identity information on employees and clients which carries judicial legalities that could be just as damaging to the company. When recovery is taken into consideration, personal and financial information might already be in the wrong hands. The damage to everyone involved would be tremendous and very time consuming in an endeavour to repair. A rule of thumb is when creating a disaster recovery plan, implement a recovery plan compatible with the business needs

Summary

You have learnt in this unit:

that some business applications cannot tolerate any downtime and only larger companies can afford dual data centres capable of handling all data processing needs, which run in parallel with data mirrored or synchronized between the two very expensive centres. Yet, backup and restore operations occur within the context of a recovery model. Thus, necessitating a recovery model; a database

property that controls how the transaction log is managed. In response to the changing threat landscape, Network Intrusion Prevention Systems were developed to provide advanced protection beyond that offered by firewalls and Intrusion Detection Systems (IDS).

Self-Assessment Questions

1. What is a recovery model?
2. Describe the Stateful Protocol Analysis Detection

Tutor Marked Assessment

- Explain the classifications of IPS

References

Retrieved on 12th October from http://encyclopedia.thefreedictionary.com/Intrusion-prevention system.

Retrieved on 12th October from https://www.ready.gov/business/implementation/ITOfficial website of the Department of Homeland Security.

Retrieved on 12th October from http://searchstorage.techtarget.com/definition/Business-Continuity-and-Disaster-Recovery.

Further Reading

- https://en.wikipedia.org/wiki/Disaster_recovery#Further_reading
- http://docs.oracle.com/cd/B19306_01/backup.102/b14192/intro001.htm

Picture of:
A Data Mamagement Facility
Photo by:
Christina Morrilo on Pexels

Module 4

File management system and Data management facilities

UNIT 1

File Management System and File System Architecture

Introduction

Having gone through previous units to have a basic knowledge of the computer file processing, the aim of this unit is to introduce the file management system and file system architecture. We shall also discuss their objectives and functions.

Learning Outcomes

At the end of this unit, you should be able to:

- give a good definition of File Management System,
- list the objectives of File Management System,
- outline some minimal requirements needed from users for file management functions,
- discuss the typical operations that must be supported in a file system,
- highlight the components in this architecture in their right order, and
- differentiate between each component in the architecture.

Main Content

File Management System

05mins

Let me say that the *file management system*, FMS is the subsystem of an operating system that manages the data storage organization on secondary storage and provides services to processes related to their access. In this sense, it interfaces the application programs with the low-level media-I/O (e.g. disk I/O) subsystem, freeing on the application programmers from having to deal with low-level intricacies and allowing them to implement I/O using convenient data-organizational abstractions such as files and records. On the other hand, the FMS services often are the only ways through which applications can access the data stored in the files, thus achieving an encapsulation of the data themselves which can be usefully exploited for data protection, maintenance and control.

Typically, the only way that a user or application may access files is through the file management system. This relieves the user or programmer of the necessity of developing special-purpose software for each application and provides the system with a consistent, well-defined means of controlling its most important asset

Objectives of File Management System

SAQ1

I will be presenting to you the objectives of a File Management System:

1. **Data Management**. An FMS should provide data management services to applications through convenient abstractions, simplifying and making device-independent of the common operations involved in data access and modification.

2. **Generality with respect to storage devices**. The FMS data abstractions and access methods should remain unchanged irrespective of the devices involved in data storage.

3. **Validity**. An FMS should guarantee that at any given moment the stored data reflect the operations performed on them, regardless of the time delays involved in actually performing those operations. Appropriate access synchronization mechanism should be used to enforce validity when multiple accesses from independent processes are possible.

4. **Protection**. Illegal or potentially dangerous operations on the data should be controlled by the FMS, by enforcing a well defined data protection policy.

5. **Concurrency.** In multiprogramming systems, concurrent access to the data should be allowed with minimal differences to single-process access, save for access synchronization enforcement.

6. **Performance**. The above functionalities should be offered to achieve at the same time, a good compromise in terms of data access speed and data transferring rate.

File Management Functions [SAQ2]

Concerning meeting user requirements, the extent of such requirements depends on the variety of applications and the environment in which the computer system will be used. For an interactive, general purpose system, the under listed constitutes a minimal set of requirements:

a. Each user should be able to create, delete, read, write, and modify files.

b. Each user may have controlled access to other users' files.

c. Each user may control what types of accesses are allowed to the user's files.

d. Each user should be able to restructure the user's files in a form

appropriate to the problem.

e. Each user should be able to move data between files.

f. Each user should be able to back up and recover the user's files in case of damage.

g. Each user should be able to access his or her files by name rather than by numeric

File System Architecture

1 Device Drivers

At the lowest level, device drivers communicate directly with peripheral devices or their controllers or channels. A device driver is responsible for starting I/O operations on a device and processing the completion of an I/O request. For file operations, the typical devices controlled are disk and tape drives. Device drivers are usually considered to be part of the operating system.

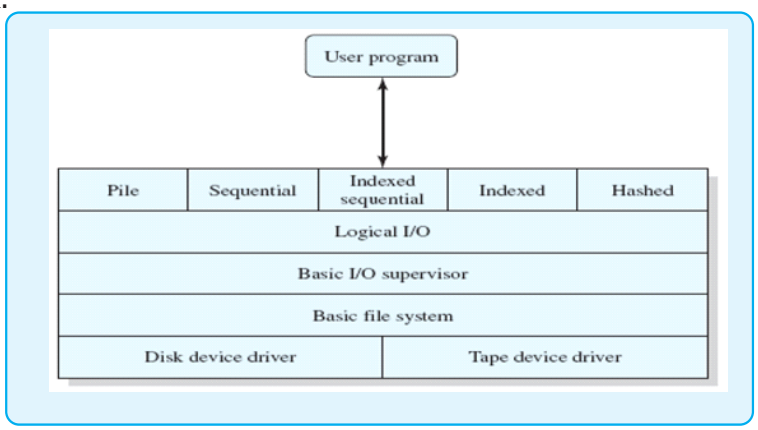


Figure 4.1:pDevice Driver

2 Basic File System

The next level is referred to as the basic file system or the physical I/O level. This is the primary interface with the environment outside of the computer system. It deals with blocks of data that are exchanged with disk or tape systems. Thus, it is concerned with the placement of those blocks on the secondary storage device and the buffering of those blocks in main memory. It does not understand the contents of the data or the structure of the files involved. The basic file system is often considered part of the operating system

3. **Basic I/O Supervisor**

The **basic I/O supervisor** is responsible for all file I/O initiation and termination. At this level, control structures are maintained that deal with device I/O, scheduling, and file status. The basic I/O supervisor selects the device on which file I/O is to be performed, based on the particular file selected. It is also concerned with scheduling disk and tape accesses to optimize performance. I/O buffers are assigned and secondary memory is allocated at this level. The basic I/O supervisor is part of the operating system.

4. **Logical I/O**

Logical I/O enables users and applications to access records. Thus, whereas the basic file system deals with blocks of data, the logical I/O module deals with file records. Logical I/O provides a general-purpose record I/O capability and maintains basic data about files. The level of the file system closest to the user is often termed the access method. It provides a standard interface between applications and the file systems and devices that hold the data. Different access methods reflect different file structures and different ways of accessing and processing the data.

Operations Supported by File Management System

[SAQ 3]

Users and applications wish to make use of files. Typical operations that must be supported include the following:

a. **Retrieve _All**

Retrieve all the records of a file. This will be required for an application that must process all of the information in the file at one time. For example, an application that produces a summary of the information in the file would need to retrieve all records. This operation is often equated with the term sequential processing because all of the records are accessed in sequence.

b. **Retrieve _One**

This requires the retrieval of just a single record. Interactive, transaction-oriented applications need this operation.

c. **Retrieve _Next**

This requires the retrieval of the record that is "next" in some logical sequence to the most recently retrieved record. Some interactive applications, such as filling in forms, may require such an operation. A program that is performing a search may also use this operation.

d **Retrieve _Previous**

Similar to Retrieve_Next, but in this case, the record that is "previous" to the currently accessed record is retrieved.

e **Insert _One**

Insert a new record into the file. It may be necessary that the new record fit into a particular position to preserve a sequencing of the file.

f **Delete_One**

Delete an existing record. Certain linkages or other data structures may need to be updated to preserve the sequencing of the file.

g **Update_One**

Retrieve a record, update one or more of its fields, and rewrite the updated record back into the file. Again, it may be necessary to preserve sequencing with this operation. If the length of the record has changed, the update operation is generally more difficult than if the length is preserved.

h **Retrieve_Few**

Retrieve a number of records. For example, an application or user may wish to retrieve all records that satisfy a certain set of criteria.

The nature of the operations that are most commonly performed on a file will influence the way the file is organized, as discussed under file organization, which in the next unit. It should be noted that not all file systems exhibit the sort of structure discussed in this subsection. On UNIX and UNIX-like systems, the basic file structure is just a stream of bytes. For example, a C program is stored as a file but does not have physical fields, records, and so on

Summary

In this unit, you have learned :

that file management system is the subsystem of an operating system that manages the data storage organization on secondary storage and provides services to processes related to their access. In this sense, it interfaces the application programs with the low-level media-I/O (e.g. disk I/O). Objectives of File Management System includes: data management, validity, protection, concurrency, performance

Self-Assessment Questions

1. Outline three objectives of file management system
2. What is a File management system?
3. List the operations supported by the file system
4. The level of the file system closest to the user is often termed to be?

Tutor Marked Assessment

- Explain any four Objectives of file management system that you know
- Explain the lowest level of the file architecture.
- What level in the file architecture is closest to the user programs?

References

Retrieved on 5th February 2019 from
www.tn.nic.intnhomeprojectfilesfilemgmnt

Retrieved on 5th February 2019 from file
www.cs.unc.edu~dewan242s07notesfile

Further Reading

- https://www.globodox.com/file-management system
- https://www.canto.com/file-management system

Photo by:
Brette Sayles
from Pexels

UNIT 2

Data Management Facilities

Introduction

In this unit, I will briefly describe data management which is often an aspect neglected in file processing. Your knowledge of operations performed on a file will help you achieve maximally from this unit.

Learning Outcomes

At the end of this unit, you should be able to:

- mention some data management procedures;
- give a brief description of database management system;
- explain two keyword structure of a DBMS;
- discuss the types of error encountered during data entry; and
- list four examples of DBMS.

Main Content

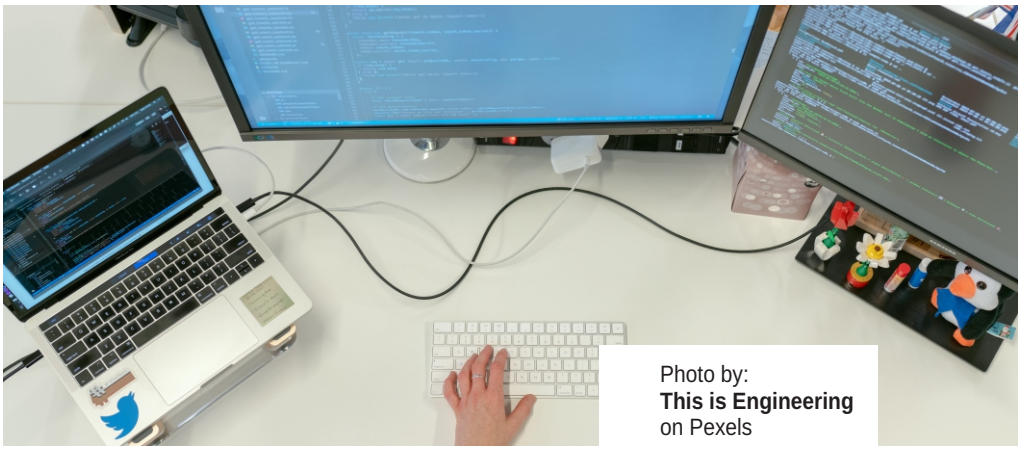


Photo by:
This is Engineering
on Pexels

Data Management

10mins

To begin with, data management includes all aspects of data planning, handling, analysis, documentation and storage, and takes place during all stages of a study. The objective is to create a reliable database containing high quality data. Data management is a too often neglected part of study design and includes:

(a) Planning the data needs of the study

(b) Data collection

(c) Data entry

(d) Data validation and checking

(e) Data manipulation

(f) Data files backup

(g) Data documentation

Each of these processes requires thought and time; each requires painstaking attention to detail.

The main element of data management is database files.

Database files contain text, numerical, images, and other data in machine-readable form. Such files should be viewed as part of a **database management systems(DBMs)** which allows for a broad range of data functions, including data entry, checking, updating, documentation, and analysis.

Data Management Software

Many DBMSs are available for personal computers. Options include:

1. Spreadsheet (e.g., Excel, SPSS datasheet)
2. Commercial database program (e.g., Oracle, Access)
3. Specialty data entry program (e.g., SPSS Data Entry Builder, EpiData)

Spreadsheet are to be avoided for all but the smallest data systems since they are unreliable and easily corrupted (e.g., easy to type over, lose track of records, duplicate data, mis-enter data, and so on). Commercially available database programs are expensive, tend to be large and slow, and often lack controlled data-entry facilities.

You should also note that speciality data entry programs are ideal for data entry and storage. We use **EpiData** for this purpose because it is fast, reliable, allows for controlled data-entry, and is open-source. Use of EpiData is introduced in the accompanying lab.

Data Entry and Validation

Be informed that **Data processing errors** are errors that occur after data have been collected. Examples of data processing errors include:

1. Transpositions (e.g., 19 becomes 91 during data entry)
2. Copying errors (e.g., 0 (zero) becomes O during data entry)
3. Coding errors (e.g., a racial group gets improperly coded because of changes in the coding scheme)
4. Routing errors (e.g., the interviewer asks the wrong question or asks questions in the wrong order)
5. Consistency errors (contradictory responses, such as the reporting of a hysterectomy after the respondent has identified himself as a male)
6. Range errors (responses outside of the range of plausible answers, such as a reported age of 290)

To prevent such errors, you must identify the stage at which they occur and correct the problem.

Methods to prevent data entry errors [SAQ 1,6]

1. Manual checks during data collection (e.g., checks for completeness, handwriting legibility)
2. Range and consistency checking during data entry (e.g., preventing impossible results, such as ages greater than 110)
3. Double entry and validation following data entry
4. Data analysis screening for outliers during data analysis

EpiData provides a range and consistency checking program and allows for double entry and validation, as demonstrated in the accompanying lab

Database Management System [SAQ 2]

Be informed that a DBMS (Database Management System) is a collection of programs that enables users to create and maintain a database.

A DBMS is hence a General-Purposes/w system that facilitates the process of databases for various applications.

There are several ways database management (DBM) has affected the field technology.

Because organizations demand directory services which have grown as they expand in size, business use directory services that provide prompted searches for company information. Mobile devices can store more than just the contact information of users and can discover (cache)and display a large amount of information on smaller displays. Search engine queries can locate data within the world wide web (www). Retailers have also benefited from the development with data ware-housing, recording, recording customer transactions. Online transactions have become tremendously popular for e-business. Consumers and businesses can make payments secondly through some company website

Five Key Words to Describe DBM

DBM systems are designed to use these keywords structure to provide simplistic access to information to the user.

The keyword structures are:

1) Network

2) Event monitoring

3) Security

4) Backup

5) Computation

Networking: The networking consists of more complex relationships; it can relate to many records and access them by following one of several paths. In other words, this structure allows and enable the user to be connected and has link relationships to work together for online effective communication. This is done through the help of database management programming.

Event monitoring: An event monitoring allows you to collect information about transient events that would be difficult to monitor through snapshots, such as deadlocks, transaction completion, and completion information that includes how long a transaction has taken place. Monitoring a database manager event results in information being returned when that event occurs. The information provides a good summary of the activities of a particular event.

Security: For security reason, it is desirable to limit who can see or change specific attributes or graphs of attribute. This may be managed directly on an individual basis, or by the assignment of individuals and privileges to groups, or in the most elaborate models, through the assignment of individuals and groups to roles which are then granted entitlements. There are three (3) security levels control access to universal database management data and functions. The first security checking is authentication when the operating system verifies through a user ID and

password.

Once authentication by the operating system authorization is the next level of security where the user must be identified to Database management by using what is called SQL authorizations name in other words "Authid". Similarly, this is what "American's security agency used to trapped down Osama bi-ladi in his hide-out". The authid can also be the same as the user ID and is normally used for proper identification at any moments in the surroundings and searching view. In essence, privileges are rights granted to users to work with objects within a database, such as a view object or searchlight.

Back-Up: Copies of attribute need to be made regular in case primary disk or the equipment fails. A period copy of attributes may also be created for a distant organization that cannot readily access the original. Database management systems usually provide utilities to facilitate the process of extracting and disseminating attribute sets. When data is replicated between databases servers, so that the information remains consistent throughout the database system and users cannot tell or even know which server in the DBMS they are using, the system is said to exhibit replication transparency.

Computation: Common computations requested on attributed are counting, summing, averaging, sorting, grouping, cross-referencing, and so on. Rather than to have each computer application implements that from scratch, which they rely on the DBMS to supply such calculations.

These given Database management optional structures depend on the natural organization of the application's data, and on the application's requirements, which include transaction rate, reliability, maintainability, scalability and cost. Database in the cause of system management, provide many facilities in addition to the control centre to aid in the management of a large, diverse database system. You can administer database client from one central location, perform database client from one central location, and perform database administration tasks remotely from a client work station unite banks for fund transfer, monitor database activity, spread databases across multiple filesystems, force users on the system, and diagnose problems.

A number of database administration management tasks can be performed while the database is still operations, "while users are still connected". This provides for greater availability of data to users. Some management tasks that can be done online include loading data, backing up data, reorganization of data, creating table spaces, and altering tables or table spaces.

Advantages of DBMS

1. The organization can exert via the DBA, a centralized management and control over the data. The database administrator is the focus of the centralized control. Any application requiring a change in the structure of a data record requires an arrangement with the DBA, who makes the necessary modification. Such modifications do not affect other applications or users of the record in question.

2. **Reduction of Redundancies:**

   Centralized control of data by the DBA avoids unnecessary duplication of data and effectively reduces the total amount of data storage required. It also eliminates the extra processing necessary to trace the required data in a large mass of data. Another advantage of avoiding duplication is the elimination of the inconsistencies that tend to be present in redundant data files. Any redundancies that exist in the DBMS are controlled and the system ensures that these multiple copies are consistent.

3. **Shared Data:**

   A database allows the sharing of data under its control by any number of application programs or users. E.g. The application for the public relations and payroll departments could share the data contained for the record type EMPLOYEE

4. **Integrity:**

   Centralized control can also ensure that adequate checks are incorporated in the DBMS to provide data integrity. Data integrity means that the data contained in the database is both accurate and consistent. Therefore, data

while the database is still operations, "while users are still connected". This provides for greater availability of data to users. Some management tasks that can be done online include loading data, backing up data, reorganization of data, creating table spaces, and altering tables or table spaces.

Disadvantages of DBMS [SAQ 3, 4]

1. Significant disadvantage of DBMS is cost.
2. In addition to the cost of purchasing or developing the software, the hardware has to be upgraded to allow for the extensive programs and workspaces required for their execution and storage.
3. The processing overhead introduced by the DBMS to implement security, integrity and sharing of the data causes a degradation of the response and throughput times.
4. An additional cost is that of migration from a traditionally separate application environment to an integrated one.
5. While centralization reduces duplication, the lack of duplication requires that the database be adequately backed up so that in the case of failure the data can be recovered. Backup and recovery operations are fairly complex in the DBMS environment. Furthermore, a database system requires a certain amount of controlled redundancies and duplication to enable access to related data items.
6. Centralization also means that the data is accessible from a single source namely the database. This increases the potential severity of security branches and the disruption of the operation of the organization because of downtimes and failures.
7. The replacement of a monolithic centralized database by a federation of independent and cooperating distributed databases resolves some of the problems resulting from failures and downtimes

values being entered for storage could be checked to ensure that they fall within a specified range and are of the correct format. E.g. The value for the age of an employee may be in the range of 16 to 65.

Another integrity check that should be incorporated in the database is to ensure that if there is a reference to a certain object, that object must exist. In the case of an automatic teller machine, for example, A user is not allowed to transfer funds from a nonexistent savings account to checking account.

5 Security

Data is of vital importance to an organization and may be confidential. Such confidential data must not be accessed by unauthorized persons.

The DBA who has the ultimate responsibility for the data in the DBMS can ensure that proper access procedures are followed, including proper authentication schemes for access to the DBMS and additional checks before permitting access to sensitive data.

Different levels of security could be implemented for various types of data and operations. The enforcement of security could be data value dependent (e.g. A manager has access to the salary details of the employees in his or her department only) as well as data-type dependent (but the manager cannot access the medical history of any employee, including those in his or her department).

6 Conflict Resolution

Since the database is under the control of DBA, he could resolve the conflicting requirements of various users and applications. In essence, the DBA chooses the best file structure and access method to get optimal performance for the response critical applications, while permitting less critical applications to continue to use the database, with a relatively slower response.

Managing Data in Table Space

Be aware that a very large database that contains large objects such as photos, or requires high performance, you need to use advance method to store your data. The database provides table spaces, containers, and buffer pools for you to define how data is store on your system. Databases are logically organized into table spaces consist of physical storage devices called containers. A Single table space can span many containers. A buffer pool is an allocation in memory used to discover(cache) table and to index data pages as they are being read from disk or being modified. You aren't required to create a table space, container, or buffer pool to create a table in a database you can accept the defaults for each when you create a database and a table in a database. By default when a database is created in a database, there default table spaces are created as follows:

Temp-space: This is a table space made temporary used to sort or reorganized tables, create indexes, and join tables

User-space: This is a regulation space used to store the tables data and indexes.

Syscat-space: A regular table space used to store the system catalog tables. Using table spaces to store your data gives you the flexibility to assign portions of a table such as data, indexes and longfield data to different table spaces. This allows you to assign different storage devices depending on the content of each table space. Table spaces can also be backed up and restored as a unit. If you separate into spaces according to back-up the table space containing the more frequently updated data more often.

Steps to Create Table

1. Start the control centre
2. Expand the folders until you see the CDLIB database.
3. Right-click the table spaces folder.

4. Select create/table space using wizard from the pop-up menu.

5. The create table space wizard appears.

Two Ways of Configuring Logging for Database Management [SAQ 5]

1. Circular logging
2. Archive logging

Circular logging: it is said to be only full, offline backups of the database are allowed to recognize.

The database must be offline i.e. inaccessible to users "when a full backup is taken. As the name suggests, circular logging uses a ring" of online logs to crashes. The logs are used and retained only to the point of ensuring the integrity of current transactions only crash recovery and version recovery are supported using this type of logging.

Archive logging: is the support recoverable database by archiving logs after they have been written. That is to say, log files are not reused. Archive logging is used specifically for roll – forward recovering. This enables the log to retain and/ or the use-exit database configuration parameter results in archiving logging. To archive logs, you can choose to have database leave the log files in the active path and then manually archive them, or you can install a user exit program to automate the archiving. Archived logs are logs that were active but are no longer required for crash recovery.

Log files can be characterized as one of the following Active-the log files written by DBM, supported crash recovery. They certain there in the files, information related to the units of works that have not.

Summary

You have learnt about the data management which is known to include all aspects of data planning, handling, analysis, documentation and storage, and takes place during all stages of a study. You also learnt its advantages and disadvantages. Types of software used for database management activities was also described together with the methods to prevent data entry errors.

Database that are very large, contain large objects such as photos or require high performance, you need to use advance method to store your data. Database provides table spaces, containers, and buffer pools for you to define how data is store on your system. Two ways of configuring logging for the database are Circular logging and Archive logging.

Self-Assessment Questions

1. State methods that can be used to prevent data entry errors
2. What is DBMS?
3. State 2 disadvantages of DBMS.
4. Explain any of the above mentioned
5. Mention two Ways of Configuring Logging for Database Management.

Tutor Marked Assignment

- Explain why EpiData is used in DBMS
- Describe in detail any two Database Management System you know.

References

- Olowu, T. C.DATABASE MANAGEMENT (DBM). Proceedings of the 1st International Technology, Education and Environment
- Conference(c) African Society for Scientific Research (ASSR)

Bennett, S., Myatt, M., Jolley, D., &Radalowicz, A. (2001). Data Management for Surveys and Trials.

Further Reading

- http://www.epidata.dk/downloads/dmepidata.pdf.
- http://repository.essex.ac.uk/2398/1/TrainingResourcesPack.pdf