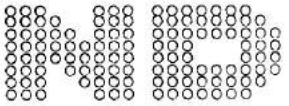


Norsk Data A.S
Postbox 4, Lindeberg Gård
OSLO 10, N O R W A Y

S I N T R A N I V

Product Description

```
============================================================
!                                                          !
!        V E R Y   I M P O R T A N T   N O T I C E         !
!        =========================================         !
!                                                          !
!    Everything contained herein may be, false,            !
!    untrue and/or directly misleading. A compound of      !
!    lies containing innumerous terminological             !
!    inexactitudes. The author will assume absolutely      !
!    no responsibility for this document until this        !
!    note is removed !                                     !
!                                                          !
============================================================
```

tim stevens

12.05.1981

TABLE OF CONTENTS

SINTRAN IV - QUALITIES
Introduction

1. QUALITIES

SINTRAN IV is both NORSK DATA's concept for and their implementation of an operating system for current and future ND computers. It is developed for a future in which independence of geographical location regarding both machines and users will be a basic demand for computer systems. The future computer system must also be able to give increased performance by merely "hooking-on" more processor power. It is a system which is open-ended and contains a very large potential for expansion and effectivisation. SINTRAN IV will be developed stepwise through a series of levels.

SINTRAN IV is built upon many foundations. NORSK DATA's experience with the well proven SINTRAN III operating system is one. Another is our knowledge of the desires and requirements of our customers, and what they believe will be important in the future. NORSK DATA's constant research into the "State of the Art" of both hardware and software development is yet another. The qualities we have aimed at in SINTRAN IV are :

- Security and Reliability
- User Friendliness
- Flexibility
- Efficiency
- Maintainability

SINTRAN IV is currently an operating system for the ND-100 and ND-500 range of computers manufactured by NORSK DATA. The future of dataprocessing will be one of distributed, decentralised processing power, so SINTRAN IV can handle :

- Single ND-500 and ND-100 systems
- Multiple ND-100 and multiple ND-500 systems
- Mixed ND-100 and ND-500 systems

The system is designed to handle a wide range of application areas, thus aiding the design of integrated solutions to application problems, in particular :

- Scientific and Technical Timesharing
- On-Line Transaction processing (Database)
- Timesharing in educational miljeau

- Text Processing and Office Automation
- Data Aquisition
- Distributed Processing and Data Communication
- Scientific and Technical Real Time (Process control)

1.1. Security & Reliability

Three different types of security are important in data processing : the security and reliability of the system itself, the security of information within the system and the reliability of individual programs.

System security

SINTRAN IV is so designed that component failure will not cause system collapse. For example, in a multi-CPU system, failure in one CPU will downgrade system performance, but the systems functionality will be unimpaired. Other failures may occur and some, e.g. a disc crash, may of course cause individual programs to halt, but the system will as far as possible continue running.

All system programs running under SINTRAN IV may be checkpointed. When a checkpoint is taken, status information for all active processes is written to a checkpoint file, and all open files are checkpointed. If a system crash occurs, the systems situation at the time of the most recent checkpoint may be reinstated, and only work done since then will be lost; exactly how much depends upon how often checkpoints are taken and on which other SINTRAN IV security features, e.g. transaction logging, have been in use.

SINTRAN IV is designed according to the "minimum priviledge principle". A user or program cannot do anything that has not been explicitly defined as permitted. The traditionnal "open system principle" works in the opposite way, and users or programs may do anything which is not explicitly prohibited.

Each user created is assigned to a USER CLASS by the system supervisor. At the time of system initialisation, each user class is given a set of PRIVILEDGES. A priviledge is a "right", allowing the owner to execute, for example, a certain monitor call or command, or to access a file in a certain manner. The user class of which the system supervisor is a member, probably the only member, will contain all the priviledges necessary for running the system itself, changing system parameters etc. Each user will be placed in a user class which allows the execution of that particular users

tasks, but no more. One priviledge is associated with each command/monitor call in the system. Thus the set of priviledges given to a user defines which subset of commands may be performed by that user.

Information security

SINTRAN IV contains many independent features which prevent unauthorised access to programs and data :

- LOGIN checking
- User Classes
- FRIENDS
- SECURITY ATTRIBUTES (to files)
- Priviledges

Each individual using the system has a USER identification, e.g. his name.

Each user may define a secret password; without knowledge of this password, it will be impossible for anyone to LOGIN, i.e. obtain access to the system via a terminal, using his user name.

Users may own FILES and DIRECTORIES. A directory is a set of files and an index containing information about their physical positions on a given mass-storage device, about their owners and who may access them. Directories and files are handle similarly by SINTRAN IV. In this document, the word "directory" will usually mean either directory or file.

Special LOGIN procedures may be specified for all or some of the user classes; such procedures may contain additional questions to establish identity, passwords etc.

The user classes mentioned in the previous section may be used to protect information from unauthorised access. The owner of each directory may define access priviledges for each user class for that particular directory. The structure of directories and files under SINTRAN IV is hierarchic. In order to access a directory at all, a user must have a READ priviledge to the directory immediately above the one to be accessed.

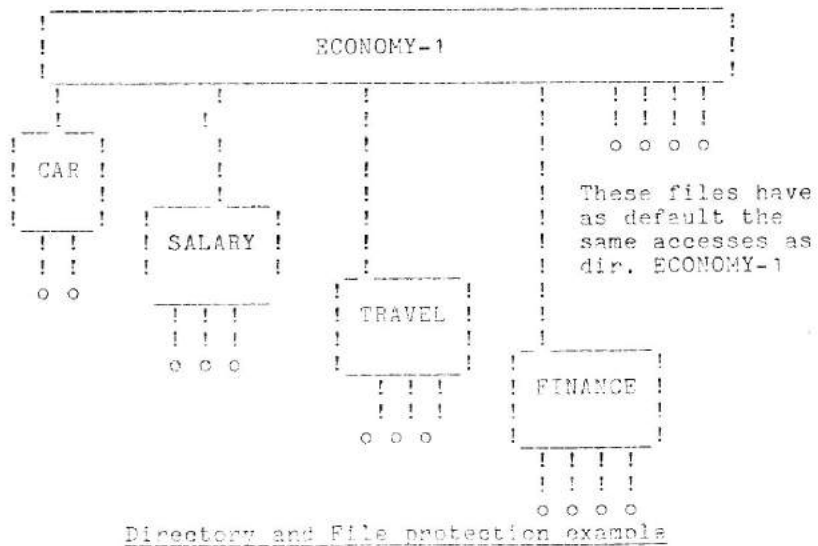
The owner of a directory may choose to allow access only to certain FRIENDS who have been explicitly defined by him. For each separate directory, the owner may define different friends, and for each friend different access priviledges. Users other than the owner and friends are known as PUBLIC users. Access by public users is controlled by each directory having a list of access priviledges for each user-class. Friends do not themselves have to be in possession of any particular

priviledges in order to access directories to which they have access priviledges as friends.

Each file has a SECURITY ATTRIBUTE which may be used both to have all acceses, or attempted accesses, to a file logged by SINTRAN IV and to prevent the copying of information to non-secure files or input/output devices. This means that an unauthorised user who suceeds in logging in with a priviledged user name will still not be able to access a SECRET file, except from a secure terminal, and will not be able to copy it except to a secure device. This facility may be used, for example, as protection against the telephonic theft of information by declaring delicate information as SECRET and defining only secure in-house devices as SECURE devices..

Access to files may be controlled by priviledges associated with PROGRAM DOMAINS (logical address spaces containing program instructions). When a user creates a program, he/she may give to that program any subset of his/her own priviledges. When a user, including its owner, runs a program, it is the programs priviledge set which determines which functions may be executed and which files accessed. Thus a users who do not themselves have sufficient priviledges to access a given file, may all the same do so by running a program which has those priviledges. For example, the accounting file may be written into only by special system routines, but these routines may be executed by all users.

Example of Directory/file protection



Directory and File protection example

Users of the system are : TRULS, MARIT, MAGNE and WENCHE

- TRULS is the financial director and is the "owner"

of directory ECONOMY-1

- TRULS sets public access to ECONOMY-1 to NONE since he wants full control over who is allowed to access it.

- TRULS declares the leaders of the different groups within the financial division (MAGNE, MARIT and WENCHE) as FRIENDS to ECONOMY-1, each with WRITE and READ access permitted.

- MAGNE, MARIT and WENCHE create, respectively, the subdirectories CAR, SALARY and TRAVEL. They are allowed to do this since they have WRITE access to parent directory ECONOMY-1.

- TRULS has created himself a subdirectory called FINANCE. He declares no FRIENDS and retains public access NONE. Nobody but himself may access FINANCE.

- MAGNE, MARIT and WENCHE declare individual FRIENDS to their directories and files. The FRIENDS in this case are people working within their groups. Each FRIEND may be given different access permissions to different directories.

Program security

SINTRAN IV was designed to assist the programmers construction of correct programs, and to trap quickly and cleanly errors which, nonetheless, occur.

Program code and data are handled separately; it is not possible to "execute" data.

SEGMENTS (Logical areas of code or data) may be protected. For example, they may be defined as "read-only" or "execute-only", thus preventing incorrect programs from corrupting data and code.

Multiple data-streams may be used, i.e. a WRITE may send information to several devices in parallel. This provides security against the breakdown of communications channels or mass-storage devices.

The checkpointing facility mentioned previously is also available for users. Both files (Delayed Update) and processes may be checkpointed, ensuring that the application system designer has available all the tools and system functions necessary for producing safe systems.

1.2. User Friendliness

The users tasks of programming, giving commands to the operating system and of running the system itself are eased in every way possible by SINTRAN IV's inherent

fault tolerance and user friendliness.

Many of the features used to enhance the user friendliness of SINTRAN IV are copied from SINTRAN III, which is renouned for its ease of use :

- All command lines may be edited
- All commands and names may be abbreviated in any unambiguous way
- A HELP command exists for every system function
- Command Parameters may be typed in directly or be prompted
- Default values exist for all parameters
- Files are easily created and manipulated

These concepts have been developed further and new ideas introduced, resulting in features like :

- A DIALOGUE PROCESSOR as the standard interface between subsystems and their users. Each subsystem has a similar command structure, has screen editing capabilities, may check parameters and provide explanatory information to the user. The dialogue processor allows parameters to be entered in any order, use default values which are displayed for the operator, save old parameter sets etc.
- A Command language with recursive, conditional and arithmetic operations (Macro language)
- A one to one correspondance between monitor calls and commands, allowing any operation which may be executed by a command to be executed inside a program and vice-versa
- A user defined file environment
- Distributed systems which are transparent to the user
- A general source code debug system with breakpoint, look-at, change facilities etc.
- Tasks, initiated from terminals, may continue to run whilst the initiating terminals are freed for other work. (Detached processes)

1.3. Flexibility

SINTRAN IV is a multi-mode operating system. That is, it is designed to run different types of tasks concurrently, e.g. Timesharing, batch, real time, transaction processing, data communication, process control etc.

The innate flexibility of SINTRAN IV's design allows different hardware configurations to be run under the same operating system. Extra discs, terminals and other devices may be accommodated as they are installed.

More processors may be added to a SINTRAN IV system in order to increase throughput without changing system function. In this case, the systems memory will be shared by the different processors, i.e. it will be a "tightly coupled" system.

Files, directories, peripherals and processors anywhere within the system may be available for use from any terminal or process within the system. They may also be protected so that only named users, or users belonging to certain user classes, may access them.

1.4. Efficiency

SINTRAN IV is designed to handle multiple processor computer systems. It consists of a hardware independent part general for all systems, and processor dependent parts for each type of physical processor present in the system. The processor dependent parts are optimised for the actual hardware being used.

Due to SINTRAN IV's efficient handling of virtual memory areas, the time required for transfering control from one process to another is minimal.

Histogram and logging functions are included to allow the tuning of the user programs, applications systems, and the operating system itself.

To ease the implementation of user software which is to run in a uniform manner on different types of devices, the Virtual Terminal Manager (VTM) and the Dialogue Processor (DP) are standard in SINTRAN IV. The Virtual Terminal Manager provides a standard software terminal interface. Each program initiated terminal function is translated to an actual function on the real terminal being used. The Dialogue Processor provides a standard interface between the user and each subsystem. Features like these, together with NORSK DATA's editors, compilers and a general source code debugger, speed up the development of user programs and thus aid the efficient use of programming resources; often the most time-consuming and expensive part of a an EDP project.

1.5. Maintainability

PLANC, a PASCAL like programming language developed by NORSK DATA, is used throughout SINTRAN IV. PLANC is a high level systems programming language which was designed to assist the construction of correct programs and provide readable and understandable code. PLANC forces modular program structures and well-defined interfaces between modules.

Due to its modular design, SINTRAN IV, easily accomodates additional modules as the need for them occurs. The extremely well-defined interface between modules assists such extensions and modifications.

When debugging a complex system, a major problem is the isolation of the error, the location of the exact lines of code where the error occurs. Because SINTRAN IV is constructed of modules which have well defined tasks and, in addition, belong explicitly to either the machine dependent or the machine independent part, faults in the operating system itself are easily identifiable.

NORSK DATA's use of the Virtual Terminal Manager, Dialogue Processor and other similar standards eases the tasks of implementing and maintaining new devices and programs, because interfaces are well defined and because the system was designed to be device independent.

2. SYSTEM COMPONENTS

2.1. Memory Management

A DOMAIN is the entire address space of a process at a given point in its execution. A domain consists of instruction address space (PROGRAM DOMAIN) and data address space (DATA DOMAIN). Each process may use several domains in the course of its execution. Domains are protected from one another. Communication between domains may be accomplished by means of monitor calls and common SEGMENTS.

Each domain is divided into 32 segments, each of which is a file containing either instructions or data (PROGRAM SEGMENT or DATA SEGMENT). The loader places program and data units into segments and assembles segments into domains.

This system of memory management has a number of advantages over traditional systems :

- Context switching is very fast since the complete memory management structure is in real memory. No clearing and reloading of page-, segment- or other tables is necessary. Only the arithmetic registers must be saved /unsaved, and the content of a single register, which indicates the currently active process, is changed

- It is easily and efficiently implemented in hardware.

- It allows real memory to be shared in an efficient manner between different processes.

- Seen from the programmers point of view, it is a very flexible system.

2.2. Distributed Processing

The future of data processing will be one of distributed, decentralised processing power. SINTRAN IV is therefore a Distributed System operating system.

Different types of processors may be coupled together using various types of hardware connection. The processors may, for example, be connected via telephone lines using either synchronous or asynchronous transmission, via a private local line using HDLC interface. SINTRAN IV is constructed so as to be independent of the actual physical connection.

Regardless of the type of physical connection used, the communication itself is transparent to the user. He does not have to know to which processor the mass-storage device containing his files is physically connected.

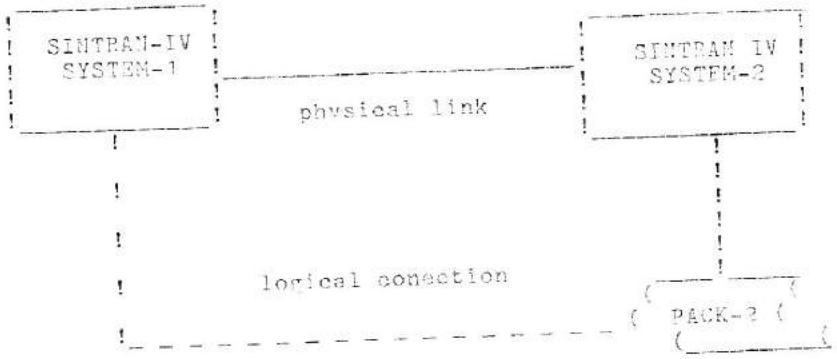
Processors may be "tightly coupled", that is they may share a common memory. On such a tightly coupled system, users need not know on which processors their jobs are executed. In other words, the user sees one SINTRAN IV system, regardless of how many processors are coupled together. Normally, tasks entered to a tightly coupled SINTRAN IV system will allow SINTRAN to decide on which processor(s) they will be executed.

If one processor in such a distributed system fails, or is to be removed from the system for maintenance or another reason,then the performance of the SINTRAN IV system will be reduced, but its functionality will remain unimpaired. User programs have facilities available to enable them to build non-stop systems inside a multiple-processor SINTRAN IV system.

Processors may also be added to a system, with the result that different CPU's may be combined to form a system with a given capacity.

Other SINTRAN IV systems (remote systems) may be addressed by name. All or part of their files may be made locally available by entering remote directories.

In the following figure, the command ENTER-DIRECTORY SYSTEM-2.PACK-TWO makes PACK-TWO available to users of SYSTEM-1



Inter-Process Communication

SINTRAN IV itself is constructed around a highly flexible and efficient Inter-Process Communications system (IPC). The same system is used for both inter- and intra- machine communication. Each process within a SINTRAN IV system may communicate with any other process within its own or any connected SINTRAN IV system. The

IPC makes the geographical situation of any process transparent to those communicationg with it.

From a user program, the IPC may be looked upon as an integral part of the I/O and file system. User systems may be constructed within a single CPU system using IPC to communicate between modules. They may then be very easily transferred to a multi-CPU system, or the original system may be expanded with more CPUs, if CPU capacity becomes a limiting factor on the systems performance. No changes will be necessary in the user programs.

The logic used in the IPC is independent of physical transmission media. The actual medium chosen may be any for which an IPC-driver exists, e.g. common memory, HDLC etc.

2.3. Timesharing

Timesharing users work from terminals and each see a virtual computer system which they may use independently of other users. Such users are allocated a share of the systems resources dependent upon the current load of the system.

From a terminal, a user may type commands to the system. Systems programs, like compilers and editors, may be run and used to compose, change, compile, test and run programs. Already existing user programs may be run.

The execution of a program may be interrupted from the terminal. The program may be aborted, halted, restarted. If a program is halted, the user may obtain from the system information about its current status.

A general source code debug system allows users to examine and change locations and to set breakpoints in their programs.

Logging facilities include a histogram, showing where a program spends its processing time, and a log showing the amounts of data transferred at each IO access.

For simple applications, only a small number of the available features will be necessary. Thus a new user, with only a minimum knowledge of the system, may quickly start to run real jobs.

For complex applications, a lot of powerful commands and features are present. For example, both conditional and loop control statements are included in the command language. The command processor also allows users to construct their own commands (macroes). These user commands may consist of combinations of other system or user-defined commands. There is a one to one

correspondance between commands and monitor calls, so that any supported language can in fact be used as a job control language.

For special applications, environments which differ from the default timesharing environment may be defined.

Each domain is divided into up to 32 segments. A file may be mapped into the address space as a segment. For example, a data file may be processed by a user program as if it was an array.

2.4. Batch Processing

A SINTRAN IV user may submit jobs to the Batch System. Such jobs run independently of other activities and without interacting with a terminal. The initiator of a batch job may be notified when the job is finished. The state of the job may be inspected and jobs submitted to a batch queue may be removed by the initiator. Batch jobs may be sent to remote systems.

The batch system contains one or more processes which execute batch jobs. A supervisor process maintains a queue of waiting jobs and schedules these jobs for execution by the batch processes.

The operator has several means available for controlling the scheduling of batch jobs. The number of batch processes is one parameter. Priority limits may be given to the different processes, causing them to reject jobs with lower priority. Jobs may be held in a waiting queue until released by the operator etc..

2.5. File System

The file system is seen by the user as an integral part of the SINTRAN IV I/O system.

The file system allows pieces of data stored on various storage media to be named and accessed by the user in a simple and uniform manner.

A basic design goal has been to make user programs as far as possible transparent to the type of data storage media actually used for the storage of their data. As a consequence, the definition of a FILE is rather wide.

A FILE is a collection of data which may be read from, written to, or both, by the computer system. The minimum addressable and accessible unit of data is one byte. Bytes in a file may be grouped into RECORDS. The most familiar example of a file is a logically consecutive

part of a disk pack or magnetic tape reel. Other examples are a whole magnetic tape reel, a line printer, a display terminal, an internal message channel or a communication channel.

The file system is designed to keep the physical properties of the storage system from the user. However, in order to avoid prohibitive inefficiencies, the distinction between random access and sequential file storage media is visible Therefore there are two main types of files : PARTITION files and VOLUME files. Partition files are always stored on random access media (usually disc), whereas volume files are stored on a sequential access medium, e.g. magnetic tape. These differences mean that the allowed set of operations which may be performed on volume files is more limited than that for partition files.

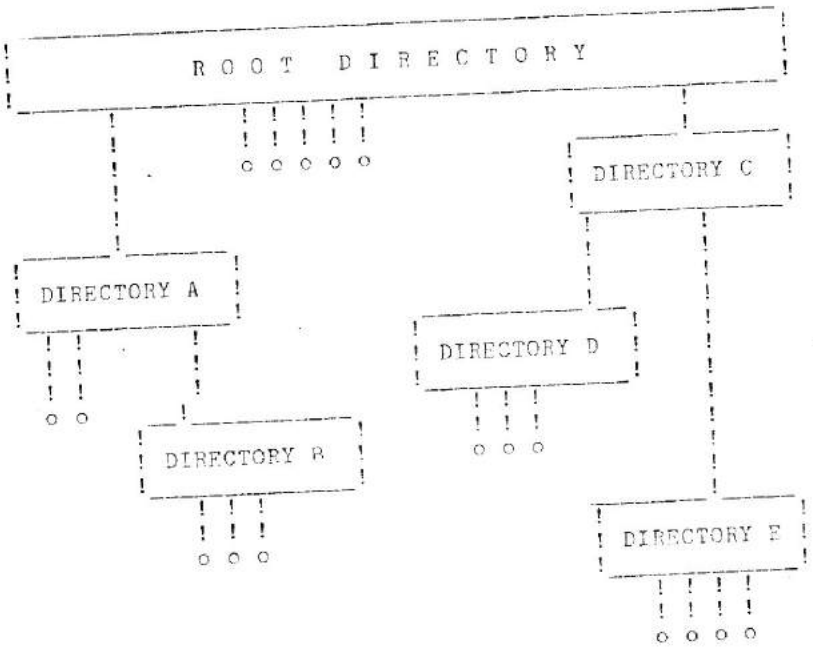
A PARTITION is the name given to the minimum unit of random- access storage that can be removed from the system without destroying the consistency of files or directories lying on it. A partition will usually correspond to one physical unit of storage, for example a disk pack or a diskette, but may also comprise two or more physical units.

However, a partition may also be stored on a file. This form of recursion may be used, for instance, to build a floppy partition image on a disc file, or to have a partition within a file fixed in memory for very fast access.

The name VOLUME is given to the physical storage unit for sequential access mass storage devices, e.g. magnetic tape. Apart from the storage format, the difference between a VOLUME and a PARTITION is that for a volume, the volume name is also a directory name. Only sequential access is allowed to files stored in volumes.

A RECORD is a group of logically consecutive bytes in a file.

Above the file level, files are grouped into directories. Directories may again be grouped into higher level directories. A directory is a group of files and/or lower level directories. A file or directory belongs to one and only one directory, except for the implicit top level directory called ROOT. An example is given in the following figure.



File names

A full file name indicates its ownership, starting with the highest level directory, through a directory hierarchy and finishing with a type or a subfile name. For example :

PACK-ONE.MY-DIR.MY-FILE:SYMB

shows that the data file MY-FILE:SYMB is a member of directory MY-DIR which is in turn a member of directory PACK-ONE.

A directory does not necessarily correspond to a physical unit of storage. A physical unit of storage is called a PARTITION.

Environment definition

Specification of the full file name for every OPEN-FILE would be very inconvenient. Therefore, each user may specify a private environment, i.e. a specific directory. All names given by the user will then be searched for inside that directory. If the file is not found in this directory, a backtrace upwards from the current environment will occur, to find the first matching file name connected to one of the owner directories. For example, system files like the FORTRAN compiler will usually be directly under the highest level directory. These files will be available from all directories which do not contain files with the same

names at some lower level.

The full file name, or at least some parts of it, must be specified if the user wishes to override his/her default environment.

User names and accesses

When a user logs on, he/she must identify himself by his/her USER NAME. This user name may allow the owner certain access rights to certain directories and files, but is not connected with the identification of them, i.e. the user name is not a part of the file name. Users are divided into user classes, 0-15, where user class 15 is the most priviledged.

File directories and individual files may be protected by their being accesible only to user classes given explicit access priviledge. Each user may define access priviledges for each directory and file he/she owns, for each user class. The access priviledges are READ, WRITE APPEND, COMMON and DELETE, or combinations of these. If a certain user does not have access to a given directory, neither will that user be granted access to any file or directory which is beneath that directory in the directory hierarchy.

This mechanism will be mainly used for the protection of system files and directories, but may also be used as a protection against unauthorised access in systems which require several independent security functions.

The main file-protection mechanism is the following: each file or directory recognises three types of accessors: the OWNER, FRIENDS and PUBLIC. When each directory and file is created, its owner decides which accesses he himself will have, which accesses explicitly named friends will have and which accesses other (public) users will have. The access rights of PUBLIC users are defined by the intersection of the rights granted them by their user classes and the PUBLIC access rights defined by the file owner.

File Versions

The directory immediately above the data file level may be defined to be a "FILE VERSION DIRECTORY". All files in such a directory are considered to be versions of the same file. The files within a directory are considered to be an ordered set. When a file in a file version directory currently opened for write is closed, it is always set as the first version, and the versions below it are shifted one step up.

File Attributes

There are six types of file ATTRIBUTES :

- FILE TYPE contains information about the physical device on which the file is stored, and about how the files are organised on that device.

- STORAGE ATTRIBUTES define the physical manner of storage of the file on the device.

- ALLOCATION ATTRIBUTES describes space allocation and lifetime of the file.

- STRUCTURE ATTRIBUTES define whether or not the file is internally structured into records or subfiles.

- SPECIAL FILE ATTRIBUTES describe special file properties. These attributes are mutually exclusive.

- SECURITY ATTRIBUTES are used to control the flow of data from one file to another.

FILE TYPE

PARTITION FILES are files stored in a directory hierarchy on a partition. Such files may be accesses either randomly or sequentially.

VOLUME FILES are those stored on a volume, usually a magnetic tape. Volume files may only be accessed sequentially. The volume labels are in accord with the ANSI standard.

PERIPHERAL FILES are the names of any types of I/O devices when they are to be accesed directly.

COMMUNICATIONS CHANNELS are files used for interprocess communication.

STORAGE ATTRIBUTE

A CONTIGUOUS file is one which is allocated to a physically continuous area within its partition.

An INDEXED file may be accessed through a 0-3 level index table. An indexed file is dynamically expandible. Only partition files may be indexed: however, to allow the backup of partition files with holes on volumes, a degenerated type of indexed file is allowed on a volume. Such a file may only be written with ascending logical addresses and read in physical sequence.

ALLOCATION ATTRIBUTE

TEMPORARY FILES are "read once" files. When such files are closed, after having been open for read (not read/write), all their pages are deleted. Temporary files must be partition files.

TRANSIENT FILES may be created and used, but disappear totally as soon as they are closed. Transient files must

be partition files.

SCRATCH FILES may be created during a terminal session or batch job. These files will be deleted when the session or job is finished. Scratch files are always partition files.

PERMANENT FILES are files which, once created, exist until they are explicitly deleted.

STRUCTURE ATTRIBUTES

Two main file structures are supported : UNSTRUCTURED and RECORD STRUCTURED.

An unstructured file is a homogenous string of bytes, as far as the file system is concerned.

In record structured files, the data is grouped into records. Four standard organisation methods are supported : SERIAL, SEQUENTIAL, KEYED and RELATIVE. Variable or fixed length records may be used under each type of organisation. Except for relatively organised files, no maximum record length need be specified.

An unstructured file may be treated as a special case of a record structured file containing a single record. Unlike most file systems which require that complete records be operated upon, the SINTRAN IV file system allows records to be read or written in parts.

SERIAL files. Records are read in the same order as that in which they are written. It is possible to skip forward, but not backwards, over a specified number of records. New records may not be inserted into a serial file, existing records may not be modified. The writing of new records may commence at any point in the file, in which case any records after that point are overwritten - the last record written in any serial file is always the last record in the file.

Serial organisation is the only kind allowed on volume files.

SEQUENTIAL files. The records may be read in the order in which they were written, or in the reverse order. A specified number of records may be skipped, either forwards or backwards. Records may be inserted anywhere in the file and existing records may be modified or deleted. The length of a record to be modified may also be changed.

KEYED files. These files have all the properties of sequential files, but, in addition, have a prime key associated with each record. The order of the records in the file will be the order of the prime key values. The prime key must be unique, and may therefore be used to

access any individual record in the file.

Any number of secondary keys may be specified. Such keys need not be unique. Access using these keys will supply the records in the order of the chosen secondary key's value. If several records share a key value, the records will be supplied in the order in which they were written - i.e. the oldest record will be supplied first.

Prime keys may be variable in length, in which case they must occur first in the record. Secondary keys must have a fixed length and a fixed position in the record relative to the end of the prime key.

RELATIVE files. Records are accessed via their positions in the files, i.e. the first record will be accessed by value 1, the second by 2 and so on. A maximum size for the records must be specified when each relative file is created.

SPECIAL FILE ATTRIBUTE

NORMAL FILES have no special properties. This is the only special attribute allowed for a volume file.

RINGBUFFER FILES work in a similar manner to ringbuffers. The file has a read pointer and a write pointer. If the file becomes full, the writing program may enter a waiting state.The reading program may wait if the file becomes empty.

MULTIVERSION FILES. If a directory is defined to be a "file version directory", all files under this directory are considered to be versions of the same file. These files will have the identifiers 1, 2, 3, ... where N is the number of versions. When a multiversion file which has been opened for write is closed, or checkpointed, it is always set as version 1 and other files identifiers are incremented. When a multiversion file is opened without the version number being specified, then if the required access is WA, then the highest version is normally opened. If the required access is not WA, then version 1 is opened.

REENTRANT FILES allow several users to access the same file simultaneously. At any instant, each user sees the original file plus the changes he has made, whilst being unaffected by other users alterations. When closed, the original file remains unaltered and all modified pages are deleted. This is a very useful file type for a subsystem, e.g. a compiler, editor etc., where the code is the same for all users but each requires his own data area.

This is implemented by giving each user a private copy of the pages he modifies, whereas unaltered pages are taken from the original file.

A DELAYED UPDATE FILE is a file for which modified pages are retained once only, regardless of how many users are working on the file. This file type is very secure, since if a system crash occurs, even if it is during the closing of the file, the original version of the file is always available.

When a delayed update file is closed, all unmodified pages are transfered to the new file and the old (unmodified) versions of the modified pages are deleted. Consequently, a consistent version of the file is always guaranteed even if the system should crash during the file update. Either all or none of the modifications made between open and close/ checkpoint will be effective.

SHARED FILE VERSIONS behave like delayed update files when they are open. The directory level immediately above such a file must be a file version directory. When such a file is closed, or a checkpoint taken, neither the old nor the new versions of updated pages are deleted. The new versions of the updated pages plus the unmodified pages are set as the first file version, and the other versions shifted one up. The pages unique to the oldest version are deleted.

In this way, version 1 of the file will always represent the state of the file at the most recent checkpoint, whilst the higher versions represent the state at earlier checkpoints.

SECURITY ATTRIBUTE

Each file has a SECURITY attribute which is used to control the flow of data between files. Each file may have one of the three attributes : SECRET, CONTROLLED or PUBLIC. If a file is defined to be either SECRET or CONTROLLED, all attempts to open the file, whether sucessfull or not, will be reported to the SINTRAN IV logging system. If the file is SECRET, the file may not be copied to a non SECRET file or device, or to a file with a lower user class. A program which has a SECRET file open for read, will not be allowed to have a non SECRET file open for write. This means that it is insufficient for an unauthorised person to succeed in logging on as a priviledged user, he must also have access to a priviledged output device in order to access secret information.

File Access Method

A file may be accessed by 3 different methods : DIRECT access, UNSTRUCTURED access and RECORD access. The access method is not a file attribute, but defines how the file is to be accessed and is, in principle, independent of the file type. There are, however, some file types with inherent limitations on the access methods which may be used. The access method is set for a file after it has been opened.

DIRECT ACCESS. The data pages of a file are read or written exactly as they are stored, including any control information and free space. The user may thus decode and organise the control information in his own manner.

UNSTRUCTURED ACCESS. The file is seen by the user as a contiguous; unstructured string of data bytes. All control information is hidden from the user and boundaries between records are invisible to him. For unstructured files, direct access and unstructured access are equivalent.

RECORD ACCESS. The file is accessed recordwise. Record control information is written and reported via parameters to monitor calls and is not seen as part of the record content. This access method is illegal on unstructured files.

2.6. Real Time

Real time tasks have the same facilities available as timesharing programs; the main difference between these two types of tasks is that real time programs have a more explicit control over their resources than do timesharing programs.

Tasks may have a fixed priority, be timesliced or choose to be guaranteed a certain percentage of processor time.

Communication between tasks is compatible with file and device input/output.

A task may consist of several processes arranged in a hierarchical structure.

Each task is identified by a name, and this name is used in all forms of communication to and concerning the task.

Real time tasks may be started from terminals and then detached from them. Tasks may be started immediately, after a certain time, at a given clock time, repeatedly at intervals or by an external event.

2.7. System Supervising

There are two levels of system start, COLD START and WARM START. Any installation may have several versions of SINTRAN, and at start time may decide which to use. A cold start starts the system up as it was generated. A warm start starts the system as it was tailored for the

individual installation, and may be used after a controlled shutdown or a system crash. The system is started with a complete configuration setup. Necessary applications initialisations may be done by a predefined batch job, so the human action involved in a warm start should be merely to start up the computer.

If a power fail occurs, no information will be lost, and the system may be restarted when the power returns.

The contents of the entire physical memory may be saved. In multiprocessor systems, the processors may be started, stopped and tested individually.

The supervisor has a comprehensive command set. Measurements may be made and statistics obtained of system performance and system parameters may be changed. Users may be created and given resources. An accounting module keeps track of system use. Supervisor commands are priviledged. The priviledge structure, i.e. which users may use which commands, may be defined for each installation.

2.8. Mail system

The mail system can be used to send messages to other users.

Two classes of message are present. Normal messages will cause the user to be warned of waiting mail when he logs in or out. Urgent messages will cause an immediate message on the recipients screen if he is logged in.

Messages may be broadcast. They will then be available for all users to read until a pre-defined expirey date. Urgent broadcasts may be sent, in which case users are notified immediately of the broadcasts existence.

2.9. Accounting

An accounting system inside SINTRAN IV will produce an accounting file which may then be printed out by a subsystem or analysed by a user written program.

The log-on and log-off times and dates, together with CPU time used and I/O time for each terminal session are recorded, similarly for batch jobs.

Use of peripherals, e.g. number of disc accesses and also disc space used will also be accounted to each user.

3. COMPATIBILITY WITH SINTRAN III

NORSK DATA have a large number of customers who will want to change over to SINTRAN IV, and/or who will want to run new SINTRAN IV systems together with existing SINTRAN III systems. Consequently a policy of as total compatibility as possible with SINTRAN III has been adopted.

SINTRAN IV contains a ND-NET module to allow communication with SINTRAN III systems, i.e. :

- Remote interactive access
- Remote file access
- Data transmission

A SINTRAN III simulation module is built into SINTRAN IV. This will allow :

- Most timesharing and batch programs to be run without either reloading or recompiling
- Programs using SIBAS, NSHS and TPS to be run
- SINTRAN III directories to be used as they are (In order to use SINTRAN III diskettes and old backup disks)
- File names used in SINTRAN III monitor calls are automatically converted to SINTRAN IV names
- Real time tasks may be moved but must be reloaded.