SUMMARY

A Modular Approach to File System Design

Massachusetts Institute of Technology, Cambridge, Massachusetts.

By Stuart E. Madnick and Joseph W. Alispach.
The Hierarchical Levels

**Figure 2**

To understand the function and interplay of each level, it is crucial to define the hierarchical structure. The purpose of the concept is to explain the role of each level and its interconnection. The hierarchical levels are defined as follows:

1. **Level 1**: The basic concepts are introduced.
2. **Level 2**: The core concepts are developed, focusing on the essential components.
3. **Level 3**: Advanced concepts are introduced, expanding the understanding.
4. **Level 4**: The more complex ideas are integrated, providing a comprehensive view.

These levels are interconnected, with each level building upon the previous one. The system design process involves navigating through these levels to ensure a cohesive understanding.

**Hierarchical Modularity**

The term “modularity” is crucial in the context of system design. It refers to the modular approach, where each component is designed to be independent yet can be integrated into a larger system. This modular design facilitates easier maintenance and scalability.

**Figure 1**

In contrast, the concept of “modularity” is applied to the system design, where each module is designed to be cohesive and independent. This approach enhances system performance and efficiency.

The two concepts are basic to the overall design.
The system design model is a concept of a hierarchical design process to provide a very flexible and versatile management framework for dynamic storage management. The system design model can be seen as a flowchart of the various levels and layers of the system. It is designed to be adaptable to various types of systems and can be modified to fit the specific needs of a particular application. The model consists of six hierarchical levels: 1. Interface, 2. Device structure, 3. Process organization, 4. Basic state, 5. Access methods, and 6. Structure. Each level has specific responsibilities and components, ensuring that the system is designed to be efficient and effective. The system design model is a comprehensive approach to system design, providing a clear and structured way to approach the design process.
A file must be split into many
direct-access devices physically.

**Modules (FOMS)**

File Organization Strategy

**Figure 3:**

![Diagram of file system organization]

**Access Methods:**

1. **Logical File System (LFS)**
2. **Basic File System (BFS)**
3. **User Interfaces**

**Levels:**

1. **Level 1:** Control System (ICCS)
2. **Level 2:** Device Strategy
3. **Level 3:** FOMS (FOMS)
4. **Level 4:** Logical File System
5. **Level 5:** User Interfaces
To minimize read/write on the I/O, the physical data is transferred to the designated buffer. After the transaction is completed, the buffer is written to the secondary storage. This process is repeated for each record.

Figure 4: Mapping Virtual Memory into Physical Records

<table>
<thead>
<tr>
<th>Physical Records</th>
<th>Virtual Memory</th>
</tr>
</thead>
<tbody>
<tr>
<td>Record 2</td>
<td></td>
</tr>
<tr>
<td>Record 14</td>
<td></td>
</tr>
<tr>
<td>Record 7</td>
<td></td>
</tr>
<tr>
<td>Record 4</td>
<td></td>
</tr>
</tbody>
</table>

In case of physical records, each record is stored in a separate sector. The location of each record is determined by the Virtual Memory Address. The location of the record is then translated into the physical location using the Translation Lookaside Buffer (TLB).

In case of virtual records, the record is located in memory, and the location is then translated into the physical location. This process is repeated for each record.

Figure 5: Translation Lookaside Buffer (TLB)

<table>
<thead>
<tr>
<th>Physical Records</th>
<th>Virtual Memory</th>
</tr>
</thead>
<tbody>
<tr>
<td>Record 2</td>
<td></td>
</tr>
<tr>
<td>Record 14</td>
<td></td>
</tr>
<tr>
<td>Record 7</td>
<td></td>
</tr>
<tr>
<td>Record 4</td>
<td></td>
</tr>
</tbody>
</table>

In case of virtual records, each record is stored in a separate sector. The location of each record is determined by the Virtual Memory Address. The location of the record is then translated into the physical location using the Translation Lookaside Buffer (TLB).

In case of physical records, each record is stored in a separate sector. The location of each record is determined by the Virtual Memory Address. The location of the record is then translated into the physical location using the Translation Lookaside Buffer (TLB).
A general concept of multi-computer network architecture is presented in this diagram. The network consists of multiple nodes connected by channels, each node representing a separate computer. The network architecture is designed to support distributed processing by allowing multiple computers to share resources and communicate efficiently.

Key points:
1. To support the accessibility to a nation-wide data base.
2. To support the co-ordination of a large organization.
3. To increase the power of one I/O device (an example of the environment to which this paper can be applied).

Multi-computer networks are by no means the only means to achieve these goals. Other approaches include shared memory systems and distributed systems. The choice of which approach to use depends on the specific requirements of the application.
of the system. Of course, the

don't work, they're not used in

the context of the volume that

will be assigned to the volume

will be assigned that the

work."

meanings of meaningful terms.

and a sound, but the

words above, the

meaning of the term "voluntary"

comprehensive triggers of volume's

concerns the figures that can be

cure the problems that are

the system. In the

triggers and the volume

of course, the

and the page is used to

universe (disconnected from an

volume) as well as

an "extraneous" device and the

sensors in the independent

continuous and unpredictable

and the volume

its data, and many of the

"shape" and the volume's

device is the "shape" and the

the network since the

of the independent pro-

system's structure or the

the position of which is

and not read in the

between the I/O

is becoming common in differ-

2. Removable volumes

would be disconnected unless

more than 64 bytes on a
disk, byte-restricted since

will affect the operation of the

mechanics of the system size

attitude capability of the

by using the storage accessible

the status of the I/O and data

into the system's co-ordinate

"I NO stored memory.

I. NO stored memory.

to the peculiarities of the

systems and combinational

solutions avoiding nontrivial

"Possibility of simplifying

the number of devices as well as

and sizes of files are with

some of these, or within

"Unreliable" devices that can

computer environment described

computer environment described

problems arising in multi-

problems arising in multi-

problems arising in multi-

CP-60/70X system is provided

by the independent pro-

For small special-purpose multi-

"Intricate" devices that can be no-

above.

computer environment described

computer environment described

computer environment described

computer environment described

computer environment described

computer environment described

computer environment described

computer environment described

computer environment described

computer environment described

computer environment described

computer environment described

computer environment described
Logical File System

The primary contribution of this work is the introduction of the conception of the Logical File System (LFS) into the structure of the file system level of the operating system. The Logical File System provides a separation of the logical and physical aspects of the file system, allowing the logical structure to be manipulated independently of the physical storage device. This separation is achieved through the use of a file catalog and a directory hierarchy, which allows for the efficient management and retrieval of files.

In a traditional file system, the file management operations are performed by the file catalog. However, in a Logical File System, the file catalog is enhanced to provide additional functionality, such as the ability to perform file system operations directly on the storage device. This allows for greater efficiency and flexibility in file management.

The Logical File System also introduces the concept of file sharing, where files can be shared among multiple users or processes. This is achieved through the use of a distributed file system, where the file catalog is replicated across multiple nodes, providing redundancy and fault tolerance.

The Logical File System is also designed to support distributed computing environments, where files can be stored across multiple machines. This allows for greater scalability and performance, as files can be accessed and manipulated from any location on the network.

In summary, the Logical File System provides a powerful framework for managing files in a modern computing environment, offering greater efficiency, flexibility, and scalability compared to traditional file systems.
These problems are solved by the

use of the Volume File Descriptor

which, normalizing the

system's module to process

the File Directory in a manner that

incorporates a mechanism that

enables the system to call back

the descriptor at the beginning of

the transaction. This allows the

system to recover from errors

that may occur in the process of

reading the file. The File

Descriptor contains information about

the file, such as its name, the

directory in which it is located,

and its attributes. The File

Descriptor also contains a

pointer to the next File

Descriptor in the Chain, which

enables the system to traverse

the file's directory structure.

The Basic File System consists of

the following components:

- The File Name Table, which
  contains the names of all
  files stored on the system.
- The File Directory, which
  contains information about
  each file, such as its
  type, permissions, and
  location.
- The File Contents, which
  contain the actual data
  stored in the file.

The system reads and writes

files by accessing the File

Directory, which is organized

in a hierarchical structure.

Example Procedure to Perform Logical File System Search

Table 1

<table>
<thead>
<tr>
<th>Case 1</th>
<th>File Index = 'READ INDEX'</th>
</tr>
</thead>
<tbody>
<tr>
<td>Case 2</td>
<td>File Volume = 'READ VOLUME'</td>
</tr>
</tbody>
</table>

The program starts by

checking the File Index

and determining if it

contains the name of a

file. If so, it proceeds to

read the File Volume

and retrieves the File

Contents. If the File Index

does not contain a file

name, the program

checks the File Volume

for the name. If the name

is found, the program

reads the File Contents.

If the File Volume does not

contain the file name, the

program checks the File

Directory for the name.
CALL F05M - READ (DESCRIPTOR) FOR AVDP-DISP.

An example of a file directory structure is shown in Figure 7.

The operation of the system is as follows:

1. The user requests a file from the file system.
2. The file system looks up the file descriptor in the file directory.
3. The file system reads the data from the file located at the specified address.

The file directory structure is hierarchical, with directories containing subdirectories and files. Each file and subdirectory is represented by a file descriptor, which contains information such as the file's name, size, and location on the storage device.

In the example shown in Figure 7, the file directory structure is organized as follows:

- Root directory
  - File 1
    - Dir 1
      - File 2
    - Dir 2
      - File 3
  - File 4
  - File 5
  - File 6
  - File 7

This structure allows for efficient file access and management, as the file system can quickly locate and retrieve files based on their directory paths and descriptors.
Example of file organization strategy

Device Strategy Module

Device Strategy Module

The file organization strategy used is the direct access method. The file is organized into fixed-length records. Each record contains a header and a body. The header includes information such as record number, record length, and a check digit. The body contains the actual data.

For other possible file organization strategies, see the following sections:

- Index sequential
- Indexed sequential
- Direct access
- Hashing
- Tree structures
The function of the panic block is to generate an interrupt request (known as a "panic") that causes an

Interrupt Service Routine (ISR) to be executed.

The panic block is a part of the process control information, which includes the process ID and the current state of the process. When a panic occurs, the ISR is executed, which can result in the process being terminated or in some other corrective action being taken.

The panic block is also used to provide a mechanism for interprocess communication, which allows processes to communicate with each other.

The panic block is a fixed-size block of memory, which is used to store the process control information. The panic block is a part of the process control information, which includes the process ID and the current state of the process. When a panic occurs, the ISR is executed, which can result in the process being terminated or in some other corrective action being taken.

The panic block is also used to provide a mechanism for interprocess communication, which allows processes to communicate with each other.

The panic block is a fixed-size block of memory, which is used to store the process control information. The panic block is a part of the process control information, which includes the process ID and the current state of the process. When a panic occurs, the ISR is executed, which can result in the process being terminated or in some other corrective action being taken.

The panic block is also used to provide a mechanism for interprocess communication, which allows processes to communicate with each other.

The panic block is a fixed-size block of memory, which is used to store the process control information. The panic block is a part of the process control information, which includes the process ID and the current state of the process. When a panic occurs, the ISR is executed, which can result in the process being terminated or in some other corrective action being taken.

The panic block is also used to provide a mechanism for interprocess communication, which allows processes to communicate with each other.

The panic block is a fixed-size block of memory, which is used to store the process control information. The panic block is a part of the process control information, which includes the process ID and the current state of the process. When a panic occurs, the ISR is executed, which can result in the process being terminated or in some other corrective action being taken.

The panic block is also used to provide a mechanism for interprocess communication, which allows processes to communicate with each other.

The panic block is a fixed-size block of memory, which is used to store the process control information. The panic block is a part of the process control information, which includes the process ID and the current state of the process. When a panic occurs, the ISR is executed, which can result in the process being terminated or in some other corrective action being taken.

The panic block is also used to provide a mechanism for interprocess communication, which allows processes to communicate with each other.

The panic block is a fixed-size block of memory, which is used to store the process control information. The panic block is a part of the process control information, which includes the process ID and the current state of the process. When a panic occurs, the ISR is executed, which can result in the process being terminated or in some other corrective action being taken.

The panic block is also used to provide a mechanism for interprocess communication, which allows processes to communicate with each other.

The panic block is a fixed-size block of memory, which is used to store the process control information. The panic block is a part of the process control information, which includes the process ID and the current state of the process. When a panic occurs, the ISR is executed, which can result in the process being terminated or in some other corrective action being taken.

The panic block is also used to provide a mechanism for interprocess communication, which allows processes to communicate with each other.

The panic block is a fixed-size block of memory, which is used to store the process control information. The panic block is a part of the process control information, which includes the process ID and the current state of the process. When a panic occurs, the ISR is executed, which can result in the process being terminated or in some other corrective action being taken.

The panic block is also used to provide a mechanism for interprocess communication, which allows processes to communicate with each other.

The panic block is a fixed-size block of memory, which is used to store the process control information. The panic block is a part of the process control information, which includes the process ID and the current state of the process. When a panic occurs, the ISR is executed, which can result in the process being terminated or in some other corrective action being taken.

The panic block is also used to provide a mechanism for interprocess communication, which allows processes to communicate with each other.

The panic block is a fixed-size block of memory, which is used to store the process control information. The panic block is a part of the process control information, which includes the process ID and the current state of the process. When a panic occurs, the ISR is executed, which can result in the process being terminated or in some other corrective action being taken.

The panic block is also used to provide a mechanism for interprocess communication, which allows processes to communicate with each other.

The panic block is a fixed-size block of memory, which is used to store the process control information. The panic block is a part of the process control information, which includes the process ID and the current state of the process. When a panic occurs, the ISR is executed, which can result in the process being terminated or in some other corrective action being taken.

The panic block is also used to provide a mechanism for interprocess communication, which allows processes to communicate with each other.

The panic block is a fixed-size block of memory, which is used to store the process control information. The panic block is a part of the process control information, which includes the process ID and the current state of the process. When a panic occurs, the ISR is executed, which can result in the process being terminated or in some other corrective action being taken.

The panic block is also used to provide a mechanism for interprocess communication, which allows processes to communicate with each other.

The panic block is a fixed-size block of memory, which is used to store the process control information. The panic block is a part of the process control information, which includes the process ID and the current state of the process. When a panic occurs, the ISR is executed, which can result in the process being terminated or in some other corrective action being taken.

The panic block is also used to provide a mechanism for interprocess communication, which allows processes to communicate with each other.

The panic block is a fixed-size block of memory, which is used to store the process control information. The panic block is a part of the process control information, which includes the process ID and the current state of the process. When a panic occurs, the ISR is executed, which can result in the process being terminated or in some other corrective action being taken.

The panic block is also used to provide a mechanism for interprocess communication, which allows processes to communicate with each other.

The panic block is a fixed-size block of memory, which is used to store the process control information. The panic block is a part of the process control information, which includes the process ID and the current state of the process. When a panic occurs, the ISR is executed, which can result in the process being terminated or in some other corrective action being taken.

The panic block is also used to provide a mechanism for interprocess communication, which allows processes to communicate with each other.

The panic block is a fixed-size block of memory, which is used to store the process control information. The panic block is a part of the process control information, which includes the process ID and the current state of the process. When a panic occurs, the ISR is executed, which can result in the process being terminated or in some other corrective action being taken.

The panic block is also used to provide a mechanism for interprocess communication, which allows processes to communicate with each other.

The panic block is a fixed-size block of memory, which is used to store the process control information. The panic block is a part of the process control information, which includes the process ID and the current state of the process. When a panic occurs, the ISR is executed, which can result in the process being terminated or in some other corrective action being taken.

The panic block is also used to provide a mechanism for interprocess communication, which allows processes to communicate with each other.

The panic block is a fixed-size block of memory, which is used to store the process control information. The panic block is a part of the process control information, which includes the process ID and the current state of the process. When a panic occurs, the ISR is executed, which can result in the process being terminated or in some other corrective action being taken.

The panic block is also used to provide a mechanism for interprocess communication, which allows processes to communicate with each other.

The panic block is a fixed-size block of memory, which is used to store the process control information. The panic block is a part of the process control information, which includes the process ID and the current state of the process. When a panic occurs, the ISR is executed, which can result in the process being terminated or in some other corrective action being taken.

The panic block is also used to provide a mechanism for interprocess communication, which allows processes to communicate with each other.

The panic block is a fixed-size block of memory, which is used to store the process control information. The panic block is a part of the process control information, which includes the process ID and the current state of the process. When a panic occurs, the ISR is executed, which can result in the process being terminated or in some other corrective action being taken.

The panic block is also used to provide a mechanism for interprocess communication, which allows processes to communicate with each other.
REFERENCES


ACKNOWLEDGMENTS

The authors acknowledge the following contributors:

1. M. J. Rencher, for his invaluable assistance in the development of the system.
2. J. R. Grainger, for his valuable input and suggestions.
3. T. J. O'Donnell, for his insightful comments.
4. R. G. Brown, for his support and encouragement.
5. J. D. McIlroy, for his expertise in the field.

The authors also thank the reviewers for their constructive feedback.

The system development was supported by the National Science Foundation, under grant No. 84-22933.
AUTHORS BIOGRAPHY

Mr. Madnick studied at the Massachusetts Institute of Technology, receiving the degree of Bachelor of Science in Electrical Engineering. He later completed the degree of Master of Science in Electrical Engineering and the associated coursework for the Master of Electrical Engineering degree. He is a member of the Institute of Electrical and Electronic Engineers, and the Association for Computing Machinery. He is a former student of the title of Ph.D. in Computer Science. This paper has been included in the selected papers recently published by the author. The author's paper, Design, State-Transition, and MIT Projects: MAC.