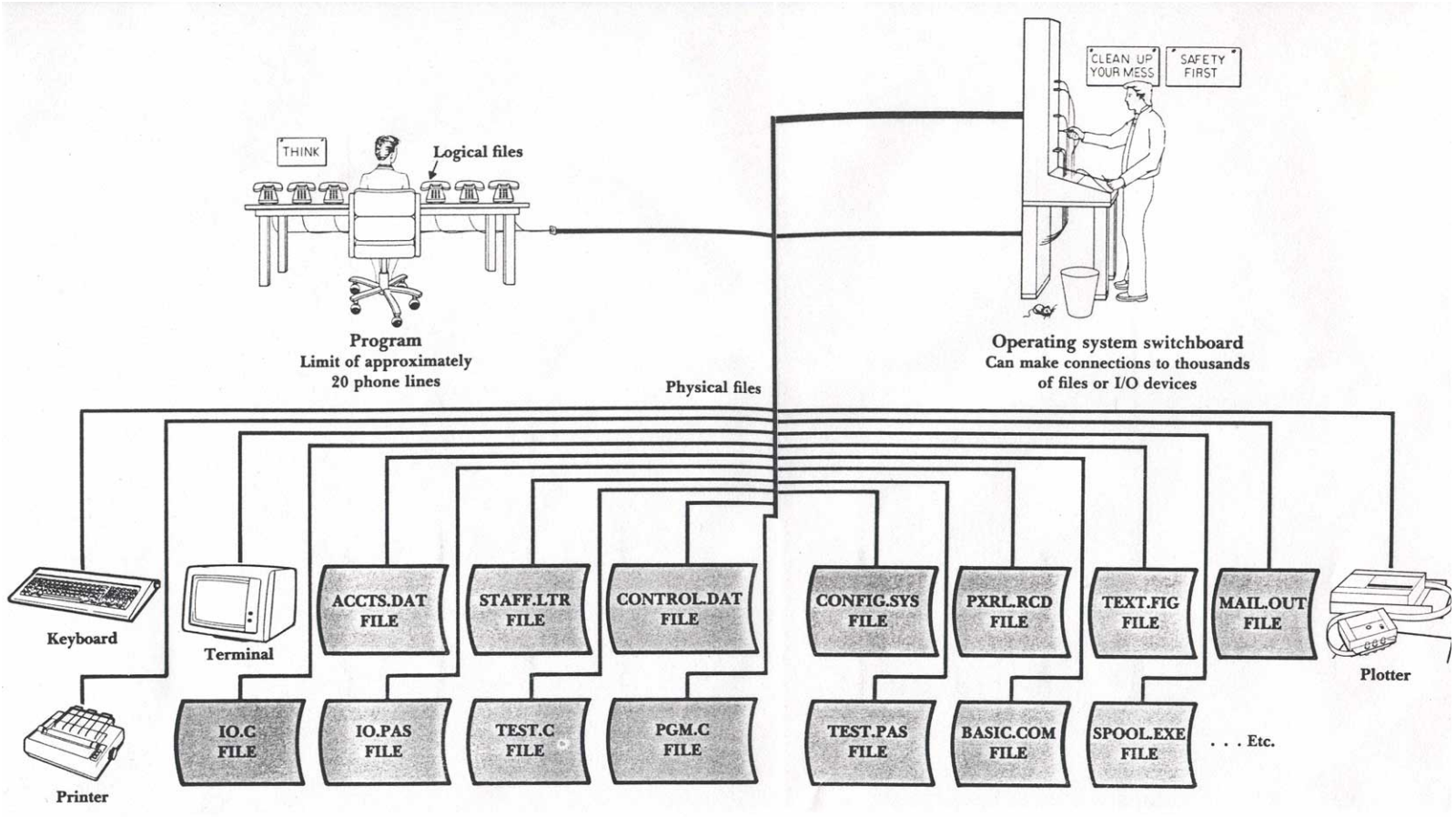
Chapter 2 :
Fundamental File Processing Operations

Physical Files and Logical Files (1)

- Physical file
  - A file that actually exists on secondary storage
  - A file in file directory that is managed by OS
- Logical file
  - A file that is seen by the program
  - Allowing the program to describe operations to be perf ormed on file not knowing what physical file will be us ed

Physical Files and Logical Files (2)



Connections between Physical files and Log ical files

- Main frame era:
  - by Job Control Language
- Unix and DOS era:
  - by the instructions within the program (O/S system call s or parts of programming languages)
  - ex) *select inp_file assign to "myfile.dat" (in COBOL)*
    - inp_file: logical file, myfile.dat: physical file
  - ex) *fd = open("myfile.dat", O_RDWR) (in C)*
    - fd: logical file, myfile.dat: physical file

System Calls vs. C Libraries for File Mana gement

- fd = open(filename, flags [, pmode])
  fp = fopen(filename, type)
- close (fd)
  fclose(fp)
- lseek(fd, offset, origin)
  fseek(fp, offset, origin)
- read (fd, buf, size)
  fread(buf, num, len, fp)
- write (fd, buf, size)
  fwrite(buf, num, len, fp)

Opening files

- Two options
  - Open an existing file
  - Create and open a file

- ex) fd = open(filename, flags [, pmode]);
  - fd: file descriptor (integer)
  - filename: physical file name
  - flags: `O_APPEND, O_CREAT, O_EXCL, O_RDONLY, O_RDWR, O_TRUNC, O_WRONLY`
  - pmode(protection mode): if `O_CREAT` is used, pmode is required
- C streams (in stdio.h)
  - file = fopen(filename, type);

Opening files: Example

- *pmode = 0751 = 111 101 001* in three-digit octal number
  - [owner, group, others] x [read, write, execute]

- Examples
  - fd = open(fname, O_RDWR|O_CREAT, 0751)
  - fd = open(fname, O_RDWR|O_CREAT|O_TRUC, 0751)
  - fd = open(fname, O_RDWR|O_CREAT|O_EXCL, 0751)

Closing files

- Closing a file
  - Making the logical file name or file descriptor available for use wit h another file
  - Ensuring that everything has been written to the file
  - ex) close(fd); (fd: file descriptor)

Reading & Writing

- Input or output operations
  - Low-level system calls (Unix)
    - read(source_file, destination_addr, size)
    - write(destination_file, source_addr, size)
  - C streams (in stdio.h)
    - fread(), fget(), fwrite(), fput(), fscanf(), fprintf()

Seeking

- Moving directly to a certain position in a file
- *lseek(source_file, offset, origin)*
  - source_file: the logical file name
  - offset: number of bytes to move from some origin in the file
  - origin: the value that specifies the starting position from which the *offset* is to be taken
- in C
  - *pos = fseek(file, byte_offset, origin)*
  - origin: `SEEK_SET, SEEK_CUR, SEEK_END`

Example Program

```
#include    <stdio.h>
#include    <fcntl.h>

main()
{
        char      c;
        int       fd;       /* file descriptor */
        char      filename[20];

        printf ("Enter the name of the file: ");
        gets (filename);
        fd = open (filename, O_RDONLY);

        lseek (fd, 501L, 0);

        while (read(fd, &c, 1) != 0)
                write (1, &c, 1);

        close (fd);
} /* end of main */
```