

**Chapter  
2**

**TASK  
DEFINITION  
IN CICS**

*Get on the  
Fast Track!*



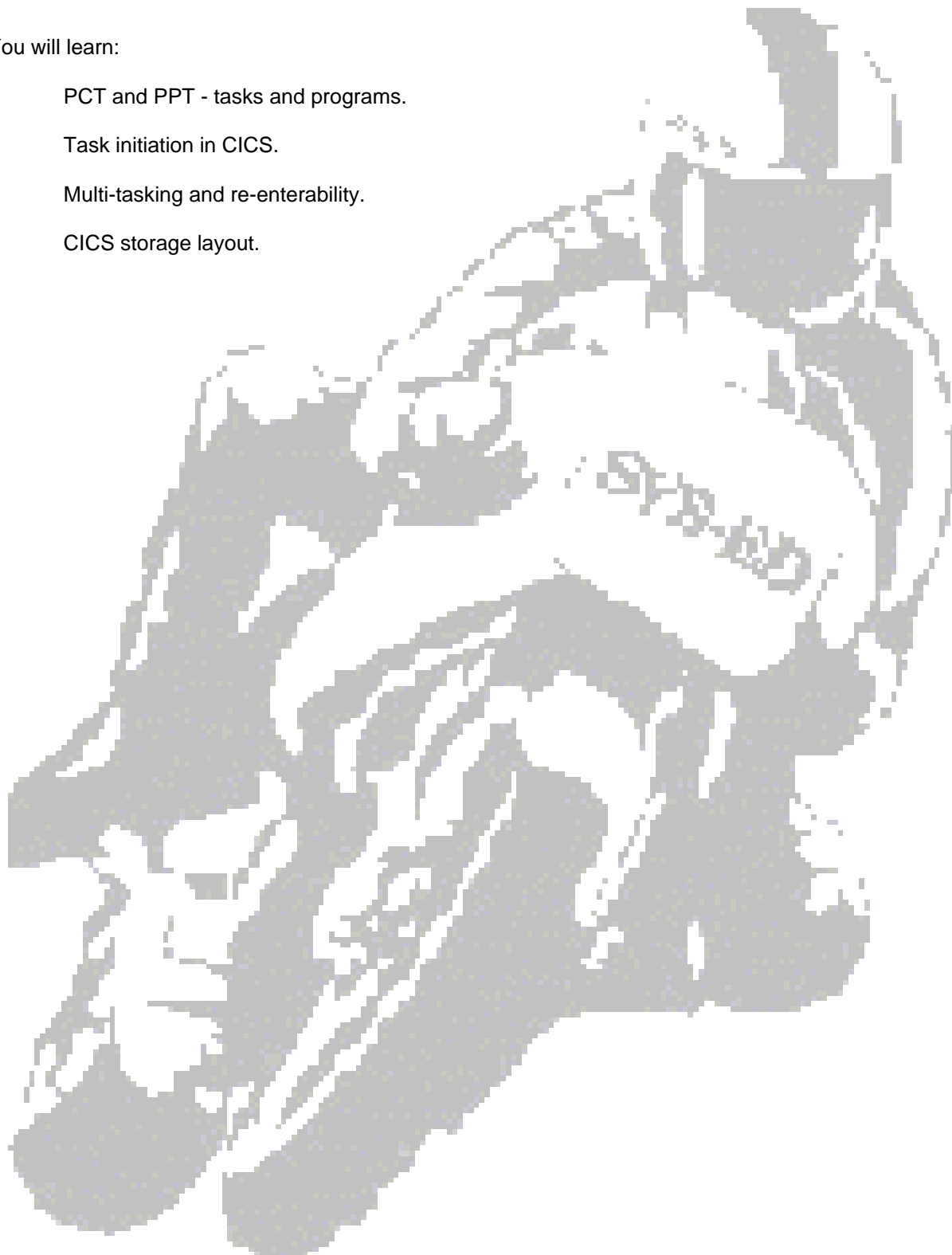
TM

**SYS-ED/  
COMPUTER  
EDUCATION  
TECHNIQUES, INC.**

**Objectives**

You will learn:

- C PCT and PPT - tasks and programs.
- C Task initiation in CICS.
- C Multi-tasking and re-enterability.
- C CICS storage layout.



---

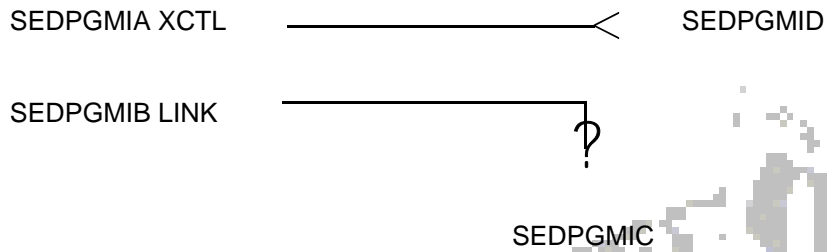
## **1 PCT and PPT - Tasks and Programs**

A task is the basic unit of system activity that is managed by CICS. The resources of the system, such as storage and CPU cycles, are allocated among all the tasks in the system, using various strategies to optimize throughput and response time.

A task is an invocation of a particular transaction. Transactions are identified to CICS by a transaction identifier (transid) of 1-4 characters, listed in the Program Control Table (PCT). A given transaction may be executed many times during the course of a day; each execution is a different task.

Each transid listed in the PCT is associated with a program. When a transaction is invoked (by specifying its transid), it is a request to CICS to execute the associated program.

The program that is directly invoked by a transid may, in turn, invoke other programs within the same task:



All application programs, those associated with trans ids, as well as other program invoked by those programs, must be listed in the Processing Program Table - PPT.

---

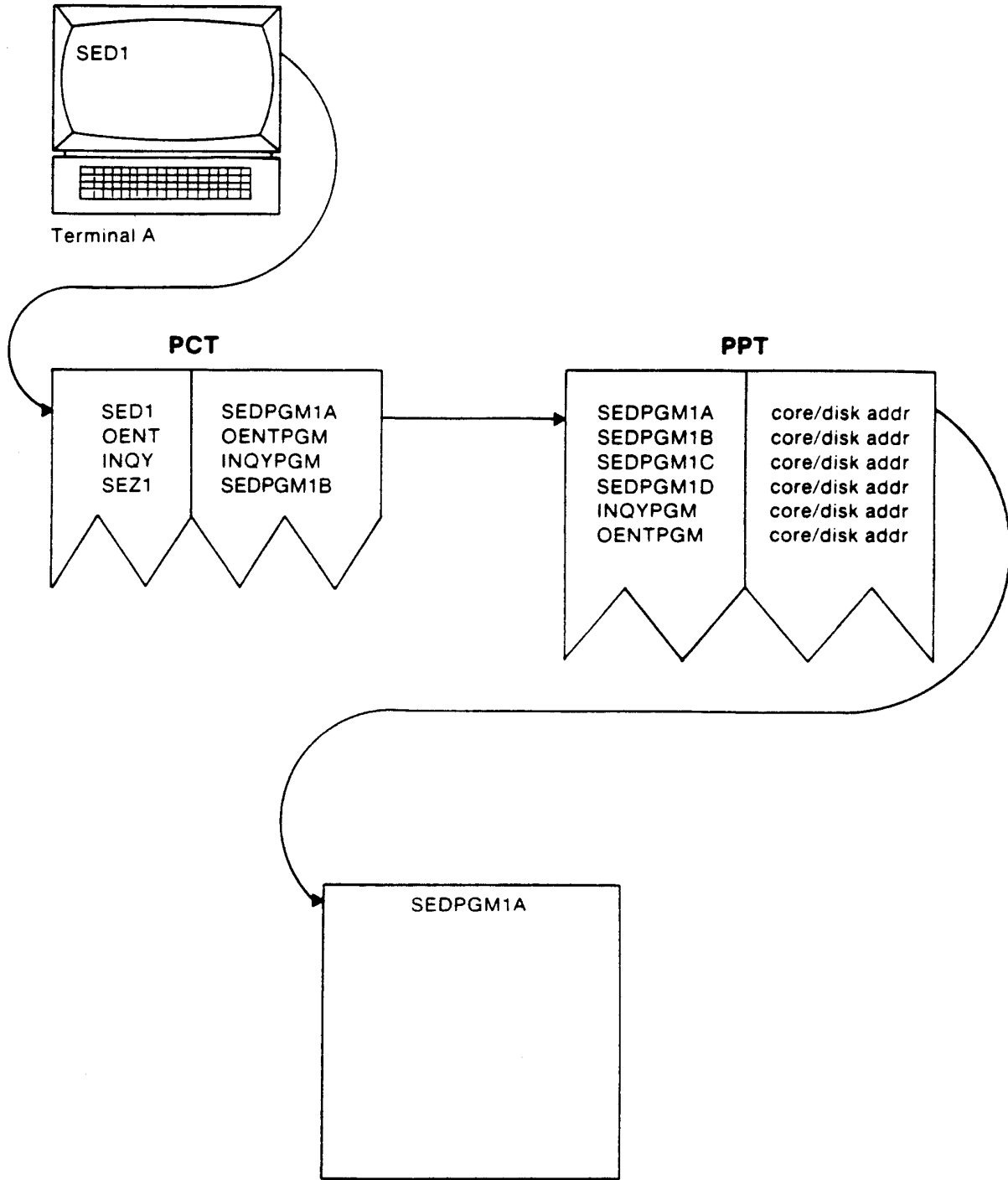
## 2 Task Initiation in CICS

Most tasks are initiated in response to a transid received by CICS from a terminal.

The sequence of events is as follows:

- 1) Operator enters transid SED1, for example, at terminal A.
- 2) CICS finds SED1 in the PCT and determines that SEDPGM1A is the specified program.
- 3) CICS looks up SEDPGM1A in the PPT and finds its core address. If the program is not in storage, it is loaded from the core image/load library and its core address is placed in the PPT.
- 4) SEDPGM1A is entered on behalf of terminal A. There is now a task attached to terminal A.

There are also other ways of initiating tasks within CICS.



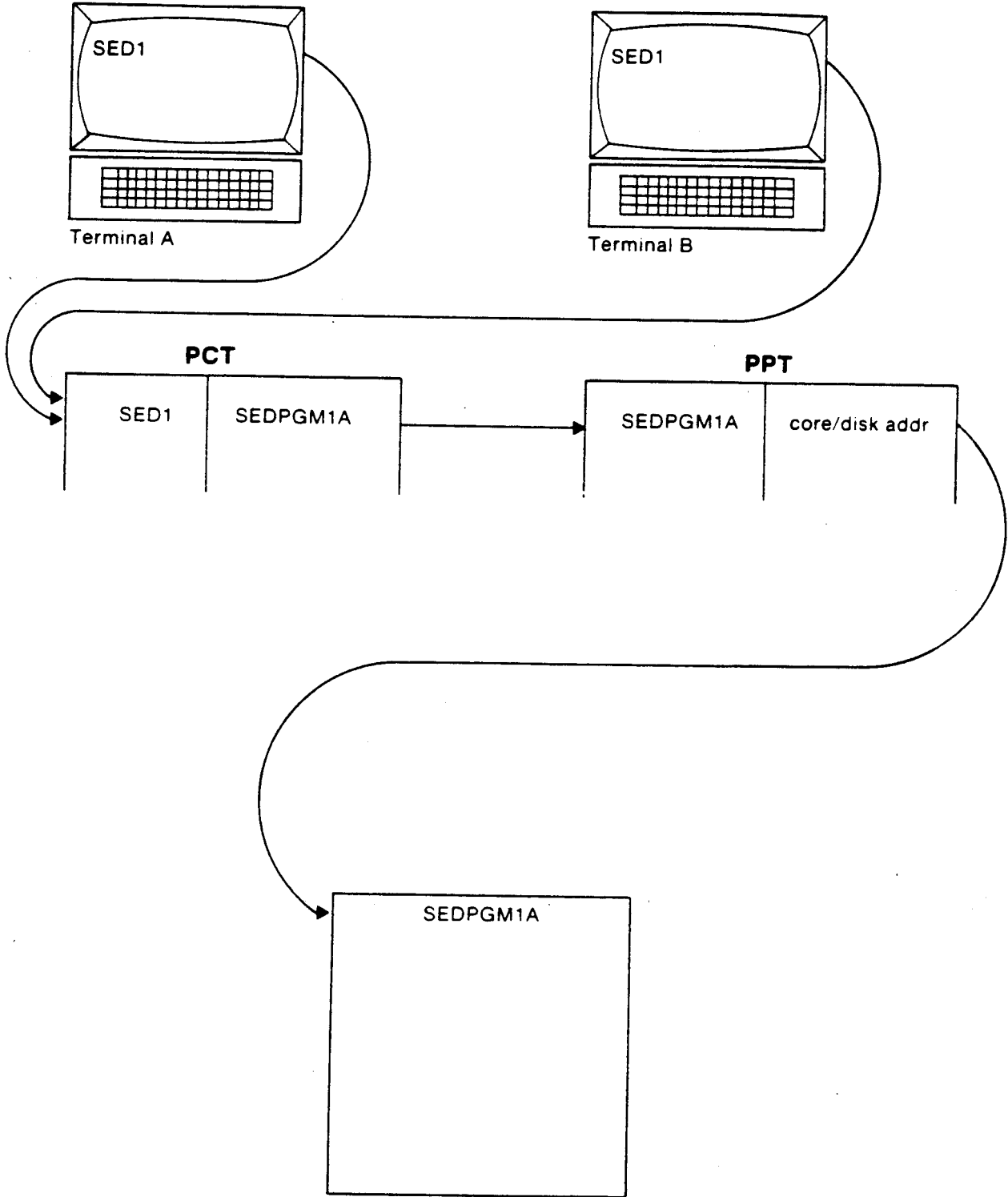
---

### **3 Multi-tasking and Re-enterability**

At any given moment, there may be one, several, or many tasks executing concurrently. The process by which CICS manages task initiation, task termination, and the allocation of system resources among the various tasks in the system, is called multi-tasking.

The same transaction may be invoked at many terminals at the same time. Each of these terminals will have a separate and distinct task attached to it, but each of these tasks (invoked by the same transid) will be executing the same program. In fact, each task will be using the same copy (in storage) of the program.

In order for the same copy of a program to be used by more than one task at a time, it must be re-entrant. This means that it must look to each task as though it is the only task using the program.



Can the following program execute on behalf of more than one terminal (task) at the same time?

```
WORKING-STORAGE SECTION
01  HOLD-REC                PIC X(100).
01  READ-SWITCH            PIC 9          VALUE 0.
01  CUST-NO                PIC X(3).
PROCEDURE DIVISION.
    IF READ-SWITCH = 1 PERFORM NO-READ
    ELSE MOVE 1 TO READ-SWITCH
      EXEC CICS READ DATASET(' SEDFILE')
        INTO(HOLD-REC) RIDFLD(CUST-NO)
      END-EXEC.
```

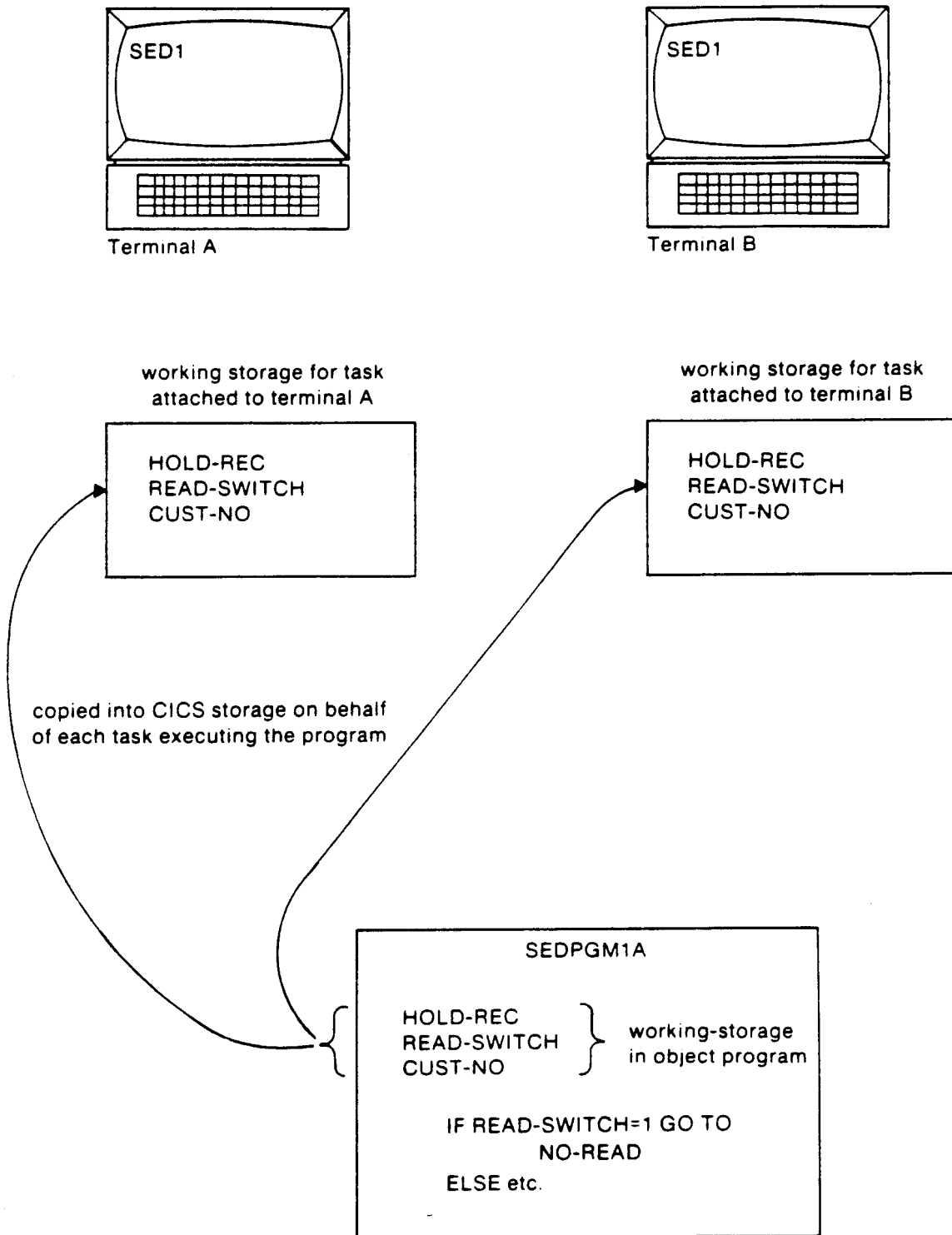
If we consider the code as written, the answer is no, because there is only one copy of the non-read-only areas such as READ-SWITCH and HOLD-REC. Thus one task may test a value of READ-SWITCH that has been set by another task. Also, each task that reads a record overlays the record being saved by another task in HOLD-REC.

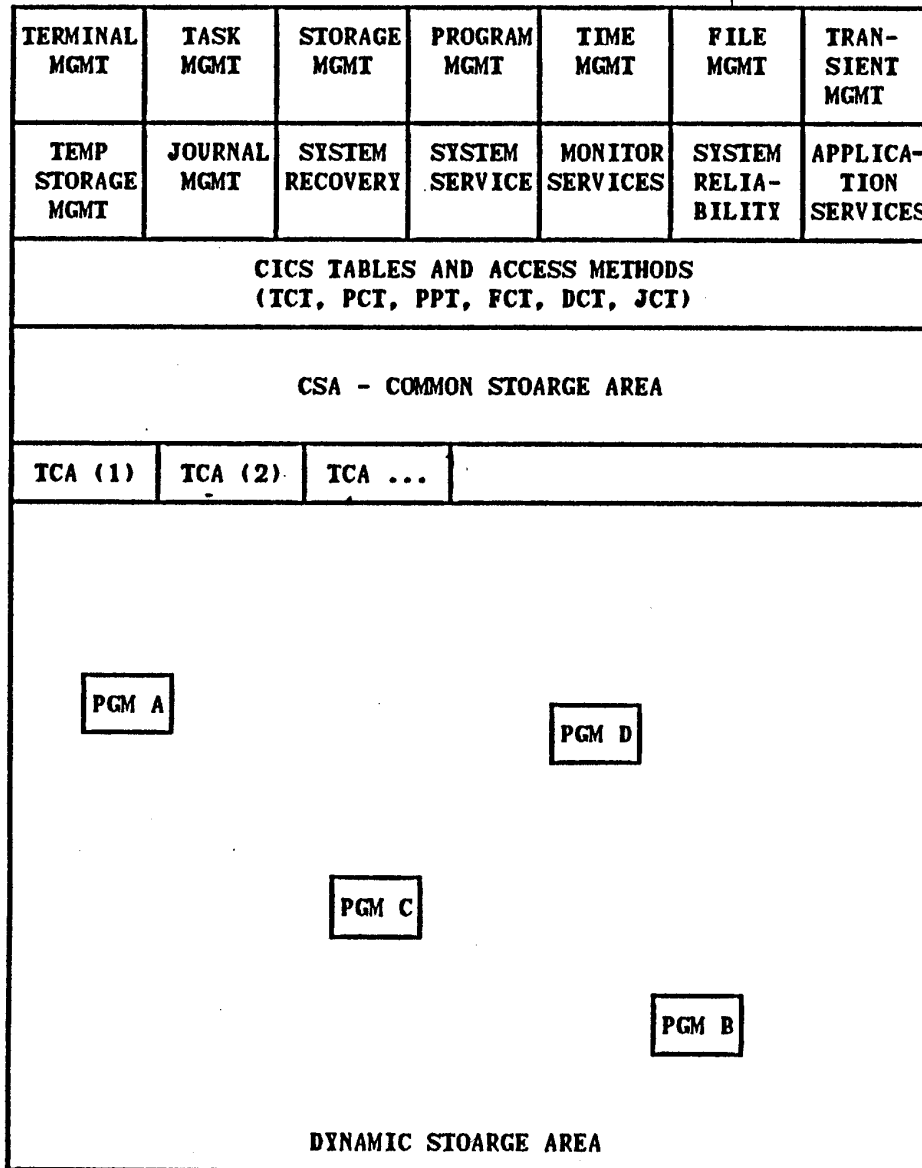
In order to achieve re-entrability, each task must have its own copy of areas like READ-SWITCH and HOLD-REC.

In fact, CICS makes a COBOL command level program re-entrant by giving each task executing the program its own copy of the working storage section.

For a PL/1 program, each task has its own copy of automatic storage.

For Assembler Language, the equivalent of working storage is defined by the programmer, using CICS macros.





- 
- 
- CICS
- MANAGEMENT
- MODULES
- 
- 

CWA, STATS,  
PIRS TO CICS  
MODS, TABLES &  
ACTIIVE TASKS

PIRS TO IWA,  
WORK-STG, TIOA  
FIOA, APP PGM