

Utilization of Available Memory on IBM 360
Computers with FORTRAN Programs

Let available memory be defined as that part of core storage to which a task has access but which is not being used by the user's routines - explicit or implied.* Six FORTRAN-compatible routines have been coded in assembly language to allow utilization of available memory and the concomitant indirect referencing of data.

Obtaining and Releasing Memory

Calling sequence: KEY = MEMGET (ADMEM, N)

MEMGET is an integer function subroutine which will attempt to obtain N consecutive full words of core storage. If sufficient available memory exists, the address of the first full word is stored in ADMEM and MEMGET will have the integer value of 1. If there isn't sufficient available memory to satisfy the request MEMGET will have the integer value of 2 and the contents of ADMEM will be unchanged.

Calling sequence: CALL MEMFRE (ADMEM, N)

MEMFRE will release to available memory the N consecutive full words which start at the memory location specified by the contents of ADMEM.

Both MEMGET and MEMFRE assume N and ADMEM are full word integers.

Determining Memory Addresses

Calling sequence: ADDA = IPTR (A)

IPTR returns an integer which is the address in core storage associated with A.

Indirect Fetching and Storing of Data

Calling sequence: CALL GETVAL (ADA, VAL)

GETVAL replaces the contents of VAL with the contents of the full word whose memory address is in ADA. ADA is assumed to be an integer.

Calling sequence: CALL SETVAL (ADA, VAL)

SETVAL places the contents of VAL into the full word whose memory address is in ADA. ADA is assumed to be an integer.

* See IBM Manual Control Program Services C28-6541-2 page 70, for more exact definition.

Indirect Transfer of Data

Calling sequence: CALL MØVMEM (AD1, AD2, N)

MØVMEM will transfer the contents of the N consecutive full words starting at the memory address contained in AD1 to the N consecutive full words starting at the memory address contained in AD2. N, AD1, and AD2 are assumed integers.

Program Characteristics

The routines are coded in assembly language and constitute a single control section of 578_{10} bytes. The MEMGET and MEMFRE routines use the GETMAIN and FREEMAIN macro and operate with sub pool 0.