Published: 11/20/68

Identification

FL (Function Language) and the Language Processor -- Overview
M. A. Meer

Purpose

Under Multics, a number of compilers and an assembler have been or are being implemented. The FL (Function Language) processor has several features not found in these or in other compilers and assemblers. The purpose of this document is to describe briefly some of the unique features of the processor and its source language.

A. Function Reference as Basic Source Program Unit

FL is designed so that each source program statement is written in function notation i.e.,
function (arguments)
This reflects the fact that each function invokes a processing unit which controls the scan of, and operates in some fashion upon, the arguments. Functions may be nested and called recursively.

B. Macro Facilities

Macro facilities exist within FL. These facilities allow the user to nest macro definitions as well as macro calls. Recursive macro calls are also permitted and by use of the PUSH and POP functions any necessary variables (symbols whose values may vary in recursive calls) may be saved.

C. Unified Symbol Table

All symbols including machine ops, pseudo-ops and register, base and modifier names are in the unified symbol table. This standard symbol table is inserted into the user's copy of the FL data segment each time FL is called (see BX.7.01). It is used and augmented by the user's program and can be output for further use in FL. This feature allows the user to define an almost completely new language for his own use. This is one of the most powerful features of FL.

Aside from programmer-written functions, the functions supplied in the standard FL symbol table are of the following types:

All of the GE-645 machine instructions. The format of the instructions conforms to standard FL input syntax, but any programmer familiar with the GE-645 machine instructions will be able to use FL functions of this type without difficulty.

Data-generating functions. Certain special functions have been added to the machine language repertoire. The DATA function interprets an argument list of data descriptors and places in appropriate locations the resulting data of given attributes. IBPAIR and ISPAIR functions generate ITB and ITS word pairs, and TALLY and TALLYD functions generate indirect tally words for indirect-then-tally address modifiers.

Directive Functions. Directive functions or pseudo-ops allow the user to communicate with and control the language processor. The IF functions for example, can be used to control the conditional assembly of source language statements. EVEN and ODD make sure that the next output code generated will fall in an even or odd location respectively. There are currently 24 such directive functions in FL.

Expression Functions. Expression functions do not stand alone but are written in the arguments of other functions; expression functions are used to:

1) Define literals and place them in a literal pool (the POOL function).

2) Assign attributes to a symbol, for example the REGISTER function.

3) Manipulate data by evaluation of expression, conversion of data types, description of variable fields, generation of ASCII text, etc., for example the OCTAL function or the ASCII function.

Multics System Standard Functions. The Multics system standard functions provide the standard interface for Multics, for example entering and returning from a procedure.

Attributes of Identifiers

Every symbol in FL has associated with it an attribute of a type such as function, location, modifier, register, etc. Attributes are generally assigned by default by by the FL processor. However, the programmer can use expression functions defined in the FL processor to override default attribute assignments. The programmer can also define his own attribute expression function.

Input and Output Streams

The FL input stream combines ease of writing with ease of comprehension. While the source program itself has a free form ASCII format, the format for each function reference within the program follows a standard format. Output from FL is also in ASCII.

FL Language Manual

The FL Language Manual will be put out as a Multics Repository Document.