Honeywell

SERIES 60 (LEVEL 6)

GCOS/BES2 Program Development

Level 6 program development extends its support to Models 6/34 and 6/36 communications, which includes a communications system configurator. Object run-time support includes a communications operator interface, a teleprinter-compatible display terminal, and an IBM BSC (2780) Host Link Support. GCOS/Basic Executive System2 program development supports the loading and activation of diskette or cartridge disk resident tasks, either as a result of another task call or as requested by the operator.

Level 6 program development is accomplished by seven stand-alone components which, in total, provide a complete program development capability. Each component is designed to run in a minimum configuration of 16K memory, use memory of up to 64K words, and make optimum use of available disk/diskette space. The seven components are:

- Command processor
- Editor
- Assembler
- Utility
- Linker
- Cross-reference program
- Macro preprocessor

COMMAND PROCESSOR

Program development jobs may be defined and prestored on cartridge disk, diskette, or card reader. The operator controls compilation and testing with the command processor, which uses the following commands:

- *Pre-execution set up:*
  - LI – Load initialize
  - AT – Attach file
  - DT – Detach file
  - PA – Print attachments
  - EX – Execute commands from command file
  - * – Command line
  - WT – Wait command
  - GO – Go command
  - IA – Initialize attach table
  - CC – Change configuration
  - IC – Initialize configuration table
  - DC – Delete configuration entry
  - PC – Print configuration table
  - QT – Terminate command file
- *Load command:* name of program to be loaded, for example, EDIT

EDITOR

Editor creates and updates FORTRAN, COBOL, or assembly language source modules on disk or diskette. Editor allows files to be edited in either batch or conversational mode, depending upon the command input stream. In batch mode, the command input stream is submitted from the card reader, disk, or diskette. In conversational mode, the command input stream is submitted from the system console.

Many powerful editing commands are provided:

- INSERT – Adds source lines
- DELETE – Deletes source lines
- PRINT – Lists all specific lines
- POSITION – FIND/LOCATE/NEXT
- SUBSTITUTE – Corrects source line(s)
- READ – Incorporates one source module within another

To edit a source file, Editor stores lines in existing memory. If more memory space is required, an intermediate file is dynamically created, utilized, and then deleted. This feature ensures that all data can be accessed in one execution of the Editor to create the final output file.

Approximately 230 eighty-character lines can be contained in memory with configurations having more than 16K memory. Single characters, words, expressions, or a string of characters can be located and changed.

ASSEMBLER

The Level 6 assembler processes source statements written in symbolic language, translating them into relocatable object code and producing a listing of the source program along with its associated assembly information. The assembler is a nonoverlaid two-pass processor which operates in 16K of main memory. During the first pass, the assembler constructs its symbol table in a memory-resident table area. During the second pass, it generates the object text and/or listing. The assembly listing reflects each source record, the corresponding machine code generated, and a maximum of four diagnostic flags per record.

UTILITY

The utility components let the user initialize new disk/diskette volumes, allocate files to them, delete and rename these files, and copy from file to file or from volume to volume. Additional components handle disk/diskette-to-memory transfer and media transcription.

Utility components available are:

- AL (Allocate) – Reserves a specified amount of disk/diskette storage space for a named file
- BTGEN (Boot Generator) – Allows users to alter the bootstrap record on disk/diskette or paper tape to conform to individual hardware configurations
- CM (Compare Data) – Checks the validity of a previous disk/diskette copy operation
- CP (Copy) – Copies data from one unit of disk/diskette storage to another
- DEBUG – Provides a flexible tool for program testing and error correction during program development
- DL (Delete) – Deletes a previously allocated file or member from a specified volume or file
- DP (Dump) – Transfers data from disk/diskette to memory or from memory to diskette or a hard-copy device
- DPEDIT (Dump Edit) – Outputs a memory image to a printer from an input diskette
- IN (Initialize) – Initializes a disk/diskette or file to a format acceptable to the file system
- LD (Logical Dump) – Prints logical records from a partitioned member or relative file
- LS (List) – Displays the contents of a specified volume directory or partitioned file index
- PATCH – Allows the user to patch an object (prelink) or executable load module (postlink) stored on disk/diskette
- PD (Physical Dump) – Prints a single, contiguous block of sectors
- PP (Disk/Diskette to Paper Tape Punch) – Transfers data from a partitioned file member (normally an executable load module) on disk/diskette to the ASR paper tape punch device
- PT (Print) – Prints the contents of a relative file or a partitioned file member in ASCII representation, using the first character as a printer control character
- RN (Rename) – Changes the symbolic name assigned to a volume, file, or member
- RP (Replace) – Sets a memory location or contiguous group of memory locations to a specified value
- XF (Card or Paper Tape to Disk/Diskette) – Transfers card or paper tape data to a partitioned file member on disk/diskette

LINKER

The linker is a single-pass component which accepts object modules produced by the assembler or by FORTRAN or COBOL compilers and joins them to produce one or more load modules. The linker:

- Resolves symbolic external references
- Builds up to a 64K-word load module
- Accepts commands from the system console or a command file, disk/diskette, or card reader
- Handles multiple input files
- Provides for automatic linking
- Produces a link map (optional)

CROSS-REFERENCE PROGRAM

The cross-reference program reads an assembly language source module, finds all labels, sorts them into alphabetical order, and lists them. On the output listing, beside each label is the number of the line where the label is defined and, sequenced numerically, the number of each line that contains a reference to this label. Duplicate and undefined labels are flagged.

MACRO PREPROCESSOR

The macro preprocessor is a nonoverlaid, one-pass processor which operates in 16K words of main memory, although up to 64K will be used if available. Input is a source program with macro call(s) and macro library files. Output is an expanded source program with macro calls generated as comment statements in addition to the macro definitions which were inserted.

SYSTEM REQUIREMENTS

Minimum equipment required:

- Level 6 CP with 16K words of main memory
- 2 diskette drives or cartridge disk
- System console (KSR teleprinter, or low-cost teleprinter-compatible CRT)

Optional equipment:

- Up to 64K words of main memory
- Additional diskette or cartridge disk drives
- Line or serial printer
- Card reader

Specifications may change as design improvements are introduced.

Honeywell

Honeywell Information Systems
In the U.S.A.: 200 Smith Street, MS 486, Waltham, Massachusetts 02154
In Canada: 2025 Sheppard Avenue East, Willowdale, Ontario M2J 1W5
In Mexico: Avenida Nuevo Leon 250, Mexico 11, D.F.

16345, 5876, Printed in U.S.A.

AT78, Rev. 2