PROGRAMMING THE IBM SYSTEM/360 MODEL 20 with RPG

ALOYSE P. OBERLE

PROGRAMMING THE IBM SYSTEM/360 MODEL 20 with RPG

Aloyse P. Oberle
International Business Machines Corporation

Prentice-Hall, Inc.
Englewood Cliffs, New Jersey

C—13-730499-4
P—13-730481-1

Library of Congress Catalog Card No.: 78-118981

Current printing (last digit):

10 9 8 7 6 5 4 3 2 1

Printed in the United States of America

PRENTICE-HALL INTERNATIONAL, INC., London
PRENTICE-HALL OF AUSTRALIA, PTY. LTD., Sydney
PRENTICE-HALL OF CANADA, LTD., Toronto
PRENTICE-HALL OF INDIA PRIVATE LIMITED, New Delhi
PRENTICE-HALL OF JAPAN, INC., Tokyo

PREFACE

This book was written with the following objectives in mind:

1. To provide the reader with the ability to design a system and write the necessary programs required to implement a basic business application on the Model 20.
2. To give the reader the general understanding and broad background that will enable him to learn more advanced design and programming concepts and techniques.
3. To provide the Systems Analyst with complete information about the Model 20 and RPG so that he can effectively design systems for implementation on the Model 20.
4. To provide non-data processing personnel, who need to know, with a basic easy reference on RPG programming and the Model 20. After reading this text, the accountant or payroll clerk would be more capable of supplying necessary information to the Systems Analyst and Programmer.

In order to accomplish these objectives the text includes:

Chapter 1—A definition of common computer terms and a description of all Model 20 Input/Output devices.

Chapter 3—A complete treatise on the Systems Design function.

Chapter 7—The use of and advantages of tape and disk files.

Chapters 4 & 9—Complete sample programs.

Chapter 10—Advanced concepts and techniques.

CONTENTS

1

INTRODUCTORY CONCEPTS AND TERMINOLOGY

IBM Computer and System/360 Terminology

The data-processing and computer field has, like most other subjects, a vocabulary all its own. Many concepts that came into being with the advent of computers required that new words be invented to describe them. This new vocabulary, which will assist the reader in understanding succeeding chapters, is explained or defined below.

Byte. A byte is one position of storage or memory in the IBM System/360. Each byte can contain a number, a letter, or a special character such as a comma or period. The byte is composed of eight bits numbered 0–7. Bits 0–3 represent the zone portion of the byte and bits 4–7 are the digit portion. An example of the letter A (zone 12 and digit 1) as it would appear in memory is

$$\begin{array}{ll} \text{Letter A} \longrightarrow & \overbrace{1\,1\,0\,0}^{\overset{Z}{\downarrow}}\quad \overbrace{0\,0\,0\,1}^{\overset{D}{\downarrow}} \\ & \uparrow\uparrow\uparrow\uparrow\quad \uparrow\uparrow\uparrow\uparrow \\ \text{Bit} \longrightarrow & 0\,1\,2\,3\quad 4\,5\,6\,7 \end{array}$$

Halfword. A halfword is two bytes.

Word. A word is four bytes.

Input. Input is information in the form of files (cards, tapes, disks, and others) which is read into the computer and acted upon by the object program.

Output. Output is information in the form of files (cards, tapes, disks, and others) which is written, punched, or printed by the computer under the control of the object program.

Input/Output Units (I/O Units). Input/output units are devices such as card readers, tape and disk drives, and printers connected to the computer which are used to read information (input) into storage and to write, punch, or print information (output) from storage.

Source Program. The source program is the instructions and data definitions written by the programmer and punched into cards.

Object Program. The object program is the machine language program created as an output from the action of the compiler on the input source program.

Compiler. The compiler is an IBM written object program that reads the source program and converts it into machine language or produces an object program so that it can be executed on the computer.

Record. A record is a card or section of magnetic tape or a disk containing related information. An example of a record would be one card containing a person's employee number, hourly rate, and department.

File. A file is a batch of related records. An example would be all the cards for a company's payroll.

Resulting Indicator. A resulting indicator is a two-digit number from 01 to 99, assigned by the programmer and used to identify records read into the computer (input) and the results of calculations performed or to condition the performance of output functions. The use of resulting indicators is illustrated throughout the remainder of the text.

Control Level Indicators. The control level indicators are L0, L1 through L9, and LR. LR (last record) goes on when the last input record has been read. L1 through L9 are used to signal control breaks during the execution of a program.

A control break is the change in a field contained in the input records that calls for the printing of totals on a report or other total functions. An example of a control break would be a change in department number in the input file.

The L0 indicator is always on and has special uses that will be illustrated later in the text.

Carriage Control Tape. A carriage control tape is a circular piece of punched paper which is inserted into the printer and controls the vertical movement of the paper in the printer. The carriage control tape is punched with a hole, called channel 1, which represents the top of a report form. It is also punched at the bottom (channel 12) of the form. Other punches may also be made in channels 2–11. These can be used for additional spacing within one page of a report.

Overflow. Overflow is the point at which there is not enough room to print additional lines on the page of a report. When an overflow is sensed (channel 12 in the carriage control tape is reached) by the computer, the program causes the printer to advance the paper to the top of the next page (channel 1 in the carriage control tape) where the printing of the report is resumed. The OF indicator, which signifies overflow, can be used by the programmer to control vertical paper movement. The use of the OF indicator is demonstrated in succeeding chapters.

First Page. 1P is the first page indicator. 1P is on only at the beginning of the program and goes off as soon as the first input record is read. One of its common uses is to control the printing of heading information on the first page of a report.

Halt Indicators. H1 and H2 are halt indicators. They are used by the programmer to cause the program to halt or stop executing if an error occurs. If one of these indicators has been turned on during the processing of a record, the object program is stopped at the completion of the processing of that record.

Logical Structure of the Model 20†

The following is a description of the logical flow of data through the IBM System/360 Model 20.

†Reprinted by permission from *IBM System/360 Model 20 Disk and Tape Programming Systems Report Program Generator*, copyright © 1966 by International Business Machines Corporation.

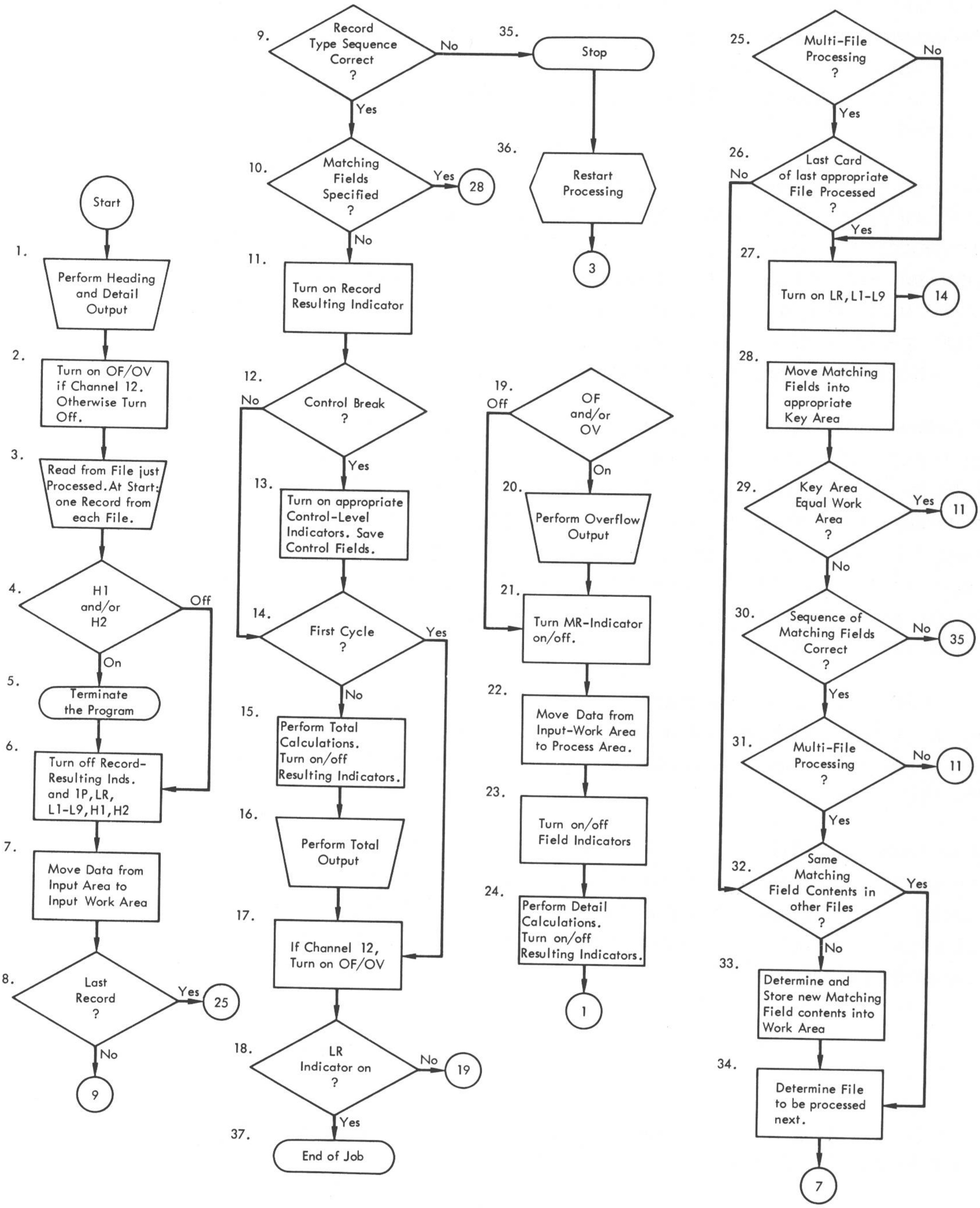
Start
1.
Perform Heading and Detail Output
2.
Turn on OF/OV if Channel 12. Otherwise Turn Off.
3.
Read from File just Processed. At Start: one Record from each File.
4.
H1 and/or H2
Off
On
5.
Terminate the Program
6.
Turn off Record-Resulting Inds. and 1P, LR, L1-L9, H1, H2
7.
Move Data from Input Area to Input Work Area
8.
Last Record ?
Yes
25
No
9
9.
Record Type Sequence Correct ?
No
Yes
10.
Matching Fields Specified ?
Yes
28
No
11.
Turn on Record Resulting Indicator
12.
Control Break ?
No
Yes
13.
Turn on appropriate Control-Level Indicators. Save Control Fields.
14.
First Cycle ?
Yes
No
15.
Perform Total Calculations. Turn on/off Resulting Indicators.
16.
Perform Total Output
17.
If Channel 12, Turn on OF/OV
18.
LR Indicator on ?
No
19
Yes
37.
End of Job
35.
Stop
36.
Restart Processing
3
19.
OF and/or OV
Off
On
20.
Perform Overflow Output
21.
Turn MR-Indicator on/off.
22.
Move Data from Input-Work Area to Process Area.
23.
Turn on/off Field Indicators
24.
Perform Detail Calculations. Turn on/off Resulting Indicators.
1
25.
Multi-File Processing ?
No
Yes
26.
Last Card of last appropriate File Processed ?
No
Yes
27.
Turn on LR, L1-L9
14
28.
Move Matching Fields into appropriate Key Area
29.
Key Area Equal Work Area ?
Yes
11
No
30.
Sequence of Matching Fields Correct ?
No
35
Yes
31.
Multi-File Processing ?
No
11
Yes
32.
Same Matching Field Contents in other Files ?
Yes
No
33.
Determine and Store new Matching Field contents into Work Area
34.
Determine File to be processed next.
7

FIG. 1–1 Courtesy of IBM.

Each object program generated by RPG [report program generator] uses the same general logic. For each record to be processed, the program goes through the same general cycle of operations. Within that cycle, there are two different instances in time when operations specified on the Calculation and Output-Format Specifications forms are performed. These instances are called detail time and total time.

A generalized flow chart of an RPG object program is shown in Fig. 1–1. The item numbers in the subsequent text refer to the numbers in Fig. 1–1. A program cycle begins with item 1 and continues through item 24.

1. The object program performs all specified heading and detail output operations whose conditions are satisfied. This does not include specifications that are conditioned by the OF or OV indicator. In addition, blank/zero indicators are turned on of those fields which have been printed, punched and/or written and are specified with Blank After. See also item 1 of *Significance of Program Logic* [which follows this section].
2. The object program performs a test to determine if a punch in channel 12 of the carriage-control tape was encountered at detail time. If this is the case, the OF/OV indicator is turned on. Otherwise, the indicator is turned off.
3. The object program makes the next input record available. At the beginning of processing, if it is a multi-file job, one input record from each file is read into core storage.
4. The object program tests the halt indicators. If the halt indicators are off, the program branches to step 6.
5. The execution of the program is discontinued.
6. All record-resulting indicators and the indicators 1P, LR, L1, through L9, H1 and H2 are turned off.
7. The data are moved from the input area to the input work area.
8. The object program performs a test to determine if the record is an end-of-file record. If an end-of-file condition has occurred, the program branches to step 25.
9. The object program performs a test to determine if the input records are in the sequence specified for them on the Input Specifications form. If the sequence is incorrect, the program branches to step 35. The program also branches to step 35 if nonsequential input records are specified and the record cannot be identified.
10. The object program branches to step 28 if matching fields are specified.
11. The resulting indicator specified for the current record type is turned on.
12. The object program performs a test to determine if a control break has occurred (i.e., the contents of this control field are not equal to the contents of a previously stored control field). If a control break has not occurred, the program branches to step 14.
13. If a control break has occurred, the appropriate control-level indicators are turned on, and the control fields are moved into a save area.
14. If this is the first program cycle, the program bypasses the total calcula-

tions and the total output specifications and branches to step 17. If no control fields are specified, total calculations and total output lines are bypassed for the first record read. If control fields are specified, total calculations and total output lines are bypassed for all records read until the first record that contains control field information has been processed. This applies also to the calculations specified with an L0 indicator.

15. All total calculations are performed and resulting indicators are turned on or off as specified.
16. Next, all total output that is *not* conditioned by an overflow indicator is performed. In addition, blank/zero indicators are turned on of those fields which have been printed, punched and/or written and specified with Blank After.
17. The object program performs a test to determine if an overflow condition has occurred (i.e., a punch in channel 12 of the carriage control tape). If an overflow condition has occurred at any time during this cycle, the indicator OF or OV is turned on.
18. The object program performs a test to determine if the last record indicator (LR) is on. If the indicator is on, the program branches to step 37.
19. The object program performs a test to determine if one or both of the overflow indicators OF and OV are on. If no overflow indicator is on, the program branches to step 21.
20. The specified overflow output is performed. In addition, blank/zero indicators are turned on of those fields which have been printed, punched and/or written and are specified with Blank After. If no overflow output is specified, the object program performs an automatic skip to a punch in channel 1 of the carriage-control tape.
21. The MR indicator is turned on if this is a multi-file job and the record to be processed is a matching record. Otherwise, the MR indicator is turned off.
22. The input data are moved from the input work area to the process area (i.e., the fields that are specified on the Input Specifications form).
23. Field indicators are turned on or off as specified.
24. Any specified detail calculations are performed, and resulting indicators are turned on or off as specified. Processing continues with step 1.
25. If only one input file is to be processed, the program continues with step 27.
26. The object program performs a test to determine if the processing of all the files specified with an E in column 17 of the File Description Specifications form has been completed. If not, the program branches to step 32.
27. All control-level indicators (L1–L9), and the last-record indicator (LR) are turned on and processing continues with step 14.
28. The specified matching fields are moved into the appropriate key area (i.e., the area that contains the control information for the file just processed).
29. Then, the contents of this key area are compared to the contents of the

matching-field work area. If the contents are equal, the program branches to step 11.

30. The program performs a test to determine if the sequence of matching fields is correct. If the sequence is incorrect, the program branches to step 35.
31. The program performs a test to determine if more than one file is to be processed. If only one input file is to be processed, the program branches to step 11.
32. The contents of the key area of the other input files are compared to the contents of the matching-field work area. If the contents are equal, the program branches to step 34.
33. Otherwise, the program determines new matching-field contents and moves the lowest or highest contents into the work area depending on whether ascending or descending sequence is specified.
34. The program determines the next file to be processed and branches to step 7.
35. The execution of the program is discontinued because of a sequence or record-type error.
36. Restart processing after elimination of the error condition.
37. The execution of the program is discontinued.

Significance of Program Logic

Four important conclusions can be drawn from the program logic:

1. At the beginning of execution of the RPG object program, all headings and detail output records, whose conditions are satisfied, are processed. At this time, the indicators L0, 1P, and all blank/zero indicators are turned on. All other indicators are turned off. The user must ensure that only the output conditions for those records that are to be printed as heading information are satisfied before an input record is read into the system. Heading information is usually specified with the indicator 1P. Other heading records must be specified with indicator N1P or' with record-resulting indicators whose conditions are not satisfied at the beginning of the object program run. As soon as the first input record is read into the system, the indicator 1P is turned off. Therefore, all records specified with the indicator 1P are not printed or punched again during the object program run. Ensure that the punching of records in a combined file is specified in such a manner that their conditions are not satisfied at the beginning of the object program run. Otherwise, the first card of a combined file is punched prior to being read. The input data contained in this card is lost.
2. When a record is read that causes a control break, the data from the record is not operated upon until after all total calculations and total output for the previous group have been performed. However, the resulting indicator specified for this record is already turned on.
3. If, during the execution of total output specifications, a punch was sensed

in channel 12 of the carriage-control tape, all remaining output specifications are executed whose conditions are satisfied. After the total lines are printed, overflow lines are printed. If a punch in channel 12 is sensed during the printing of a detail line, all the remaining detail and total lines specified (whose output conditions are satisfied) are printed before the printing of overflow lines. Therefore, the punch in channel 12 must be correctly placed to permit printing of these lines on the page.

4. If a halt indicator (H1 or H2) specified in Field Indicators on the Input Specifications form or (for detail calculations) in Resulting Indicators is turned on, the object program terminates processing immediately after completing this detail cycle. If a halt indicator is set on during the processing of total calculations, the object program continues processing until the end of the next detail cycle is reached. The execution of any specifications can be conditioned or prevented by placing H1 and H2 or their negation NH1 and NH2 in Indicators of the appropriate specifications form. This termination of the program must not be mistaken for the regular end-of-job. The desired program execution can be continued by pressing START. In this case the program continues processing as described from item 6 of Fig. 1–1.

Model 20 Input/Output Devices

This section contains a description of the I/O devices that can be connected to the Model 20.

2501 Card Reader

- Model A1 reads 600 cards per minute.
- Model A2 reads 1000 cards per minute.
- The 2501 Card Reader contains one 1200-card-capacity hopper and one 1300-card-capacity stacker.

2560 Multi-Function Card Machine (MFCM)

- Provides the combined facilities of a card reader, card punch, collator, and card interpreter document printer.
- Equipped with two hoppers (1200-card capacity each) and five stackers (1300-card capacity each). The MFCM can read or punch from either hopper.
- Any card can be directed to any of the five stackers.
- Reads 500 cards per minute from either feed (hopper) or combination of feeds.

- Punches 160 columns per second.
- Optional: Card Print Feature for printing 2, 4, or 6 lines on a card, 64 alphameric print positions per line, spaced 10 per inch. Feature is operator-adjustable to print on any of 25 lines on the card.

2520 Card Read Punch and 2520 Card Punch

- Serial reading and punching, each equipped with a 1200-card-capacity hopper and two 1300-card-capacity stackers.
- Model A1 reads and punches 500 cards per minute.
- Model A2 punches 500 cards per minute.
- Model A3 punches 300 cards per minute.

1442 Card Punch

- Punching speed depends on the last column punched and ranges from 91 cards per minute when all 80 columns are punched to 270 cards per minute when only the first 10 columns are punched.

1419 Magnetic Character Reader

- Reads magnetically inscribed data from card and paper documents at a maximum rate of 1600 documents per minute.
- Pocket selection may be controlled either by the 1419 or by the Model 20 program.
- May be used for both online and offline sorting.

1403 Printer

- Three models of this high-speed printer are capable of printing 48 standard characters or 240 characters with the Universal Character Set (UCS) feature. Line spacing of 6 or 8 lines to the inch is under operator control. Speeds vary according to the model and character set utilized.
- Model 2-132 print positions, dual-speed carriage, 600 LPM (lines per minute) standard, maximum 750 LPM with UCS.
- Model 7-120 print positions, single-speed carriage, 600 LPM, UCS not available.
- Model N1-132 print positions, dual-speed carriage, 1100 LPM standard, maximum 1400 LPM with UCS.

2203 Printer

- 120-position print span with 24 additional print position available as optional feature.
- Line printing speeds variable according to need by operator-changeable typebar.
- Model A1 Prints:
  750 LPM with 13-character set (numeric).
  425 LPM with 39-character set.
  350 LPM with 52-character set.
  300 LPM with 63-character set.
- Automatic tape-controlled carriage.
- Dual feed carriage available as optional feature.

2415 Magnetic Tape Unit and Control

- Each model has one single-channel tape control and multiple independently operated tape drives. Models 1 and 4 contain two tape drives, Models 2 and 5 each have four, and Models 3 and 6 each have six.
- Tape units utilize heavy-duty half-inch magnetic tape and a nine-track read/write head (eight bits plus parity). Recording density is at 800 or 1600 bytes per inch depending on the model.

Characteristics	*Models 1, 2, 3*	*Models 4, 5, 6*
Bytes per second transfer rate	15,000	30,000/15,000
Density (bytes per inch)	800	1600/800
Tape speed (inches per second)	18.75	18.75
Nominal IRG time (milliseconds)	32.00	32.00
Rewind and unload (minutes)	4.0	4.0

2311 Disk Storage Drives, Models 11 and 12

- The 1316 Disk Packs utilized on the 2311 have a fixed-sector format and rotate at 2400 RPM (revolutions per minute), giving a rotational period of 25 msec (milliseconds) and a read/write rate of up to 156,000 bytes per second.
- Capacity characteristics of the 2311 Models 11 and 12 are as follows:

	Model 11	*Model 12*
Bytes per sector (data only)	270	270
Sectors per track	10	10
Track capacity (data only)	2700	2700
Cylinders	200	100
Total data capacity	5.4 million bytes	2.7 million bytes