

MVS Systems Programming

Dave Elder-Vass

Reviewed by Michael S. Hines

toExcelI an imprint of iUniverse.com, Inc, 2000, 530 pp
ISBN 0-595-00184-X

MVS Systems Programming is intended to fill the gap from beginning System Programmer to productive System Administration team member. Elder-Vass indicates he wrote the book due to the gap between starting Systems Programming work and using the existing available resources at that time. His book is intended to fill that gap. His goal is take the beginning Systems Programmer through the steps necessary to install, start, maintain and enhance MVS Systems (the current technology at the time of his writing. While the material may appear dated, most of it is still relevant today – as MVS evolved into zOS). The most recent information is always available in the free IBM Documentation web site (<http://www.ibm.com/support/documentation/us/en/>).

The book begins by providing some background on the job, the hardware, and MVS internals. This sets the stage for the remainder of the book. He provides a list of 18 tasks and skills that define the Systems Programmers job. In looking at the hardware he describes each component and how that component interacts with other components to perform computing and Input/Output. He examines the control block structure to describe how MVS internals work (this is typical for most any operating system – control blocks contain the internal data that controls the system operations) to provide multi-processing, multi-tasking, and security.

In the second section he describes some of the 'tools of the trade' – basic tools such as TSO, MVS utility programs, Console commands, etc. Then a whole chapter is devoted to SMP/E processing to generate or maintain systems. Anyone who has worked with SMP/E knows that this alone could take a whole book (and in fact is a whole IBM Manual in and of itself). Lastly he talks about the need to understand Assembler programming concepts and machine instructions and operations. This helps you understand that the Assembler macro RST does not really mean Rewind and Shred Tape (a joke – in fact it may not even be a real macro!). For a good reference for JCL and MVS utility programs I recommend **System 390 JCL, 4th Ed** by Gary DeWard Brown.

The next section walks you through the process of generating a working system – MVS initialization, creating SYSRES (the system volume), configuring the I/O subsystem, JES2 initialization (you JES3 folks are on your own here), and basic network configuration – NCP or VTAM. Lastly the book helps you prepare TSO, ISPF, and ISPF/PDF for operations.

Once you have a working system you usually maintain and enhance the system with other IBM or Third Party software. This entails installing system maintenance (SMP/E again!), partitioning the system – physically and logically using PR/SM, creating multi-image systems (multiple MVS partitions running in parallel, sharing resources), and possibly sharing I/O devices.

Housekeeping chores are extremely important for the Systems Programmer. This entails MVS Security (usually an extra installed product), DASD management, and System Data housekeeping – SMF data, SYS1.LOGREC, SYSLOG, dump datasets and SYS1.PARMLIB data.

Last, the Systems Programmer becomes second level support (sometimes first level – depending on your location in the organization) for technical system problems. As such you must understand problem diagnosis and resolution, including using dump data to diagnose problems. And you always plan for the

worst by having an emergency recovery facility – be it an alternate IPLable* SYSRES, alternate MVS Systems, or standby systems available.

If you are new to IBM Mainframe Systems Programming – or even if you are an experienced Systems Programmer – there is much 'meat' in this volume that will help you do your job better. The large volume (8.5 x 11 format – 530 pages) is full of useful material to help you learn about your MVS/zOS systems so that they function most effectively.

Honestly, I never would have figured out SMP/E if it hadn't been for this reference material first, followed by the IBM SMP/E manual for current details on the software. I highly recommend this reference manual.

- IPL = Initial Program Load – aka 'booting' in other platforms.