Christian Grothoff
christian@grothoff.org

http://grothoff.org/christian/

1These slides are based in part on materials provided by IBM’s Academic Initiative.
Today

Responsibilities of a z/OS system programmer:

• Use and management of system libraries
• Configuration of consoles
• IPLing a system
What is a Systems Programming?

- A systems programmer installs, customizes and maintains the operating system

- He needs to know:
  - Hardware (Storage, processors)
  - Software (System libraries and data sets)
  - Current customization
System Programming

- System performance and workload management
- System parameters and system libraries management
- z/OS new features implementation and z/OS system maintenance
- Security, Availability and Integrity
- Controlling operating activities and functions
- Hardware I/O configuration
z/OS Administration Tasks

- Software installation and maintenance
- Maintain system libraries for software
- Manage system data sets
- Manage z/OS system address spaces and subsystems
- Manage real and virtual storage
- Monitor and improve system performance

⇒ Customize parameters
Considerations for a New Application

- New batch jobs for the scheduler
- New JCL procedures for the procedure library
- New load libraries
- Documentation of operational procedures
- Security privileges
- Reload system (if required)
- Automation
z/OS day-to-day operation

- Operator interaction is message and command based
- Batch schedules take care of repeated processes
- Automated processing of messages and commands is available

⇒ Operators manage by exception
⇒ Important task is investigation of batch failures
Change Control

Disciplined change management processes and policy enforcement helps with:

• Availability
• Security
• Audit readiness
• Cost savings

⇒ Change control is part of the job of a system programmer.
Considerations for Changes

- Benefits from the change
- What happens if the change is not done
- Resources required to implement the change
- Relative importance compared to other change requests
- Interdependencies between change requests
I/O device management

- I/O device configuration must be defined to both hardware and software
- HCD is used to build an I/O definition file
- This definition can be activated to both software and hardware dynamically
- Major changes require an IPL of software or POR of hardware
System performance

• System tuning is an iterative, ongoing process

• Initial set up of initiators and other resources plays a great part

⇒ WLM is one component to understand, monitor and configure
Types of Data

- z/OS software
- Non-z/OS (CICS, DB2)
- Non-IBM software
- Customization data
- User defined exits
- User data
Important z/OS System Libraries

Important libraries start with SYS1:

**LINKLIB** prime system software library

**LPALIB** system subroutines in LPA (more later)

**NUCLEUS** basic supervisor modules

**PROCLIB** system JCL procedures

**PARMLIB** control parameters (/etc)

**SVCLIB** supervisor call routines
PROCLIB concatenation ≠ SYS1.PROCLIB

- SYS1.PROCLIB is only one library in the overall PROCLIB concatenation
- The same applies to (SYS1.)PARMLIB and other concatenations
- Which data sets are part of the PROCLIB or PARMLIB concatenation is determined by the installation-specific z/OS configuration
Example: Linklist concatenation on Marist

IEASYS99 specifies “PROG=8W”, which means that the link list is defined dynamically in the PROG8W member:

```
APF FORMAT(DYNAMIC)
APF ADD
    DSNAME(SYS1.SHASLNKE) VOLUME(&SYSR1)
APF ADD
    DSNAME(SYS1.SIEAMIGE) VOLUME(&SYSR1)
LNKLST DEFINE NAME(LNKLST00)
LNKLST ADD NAME(LNKLST00) DSN(SYS1.MARIST.LINKLIB) VOLUME(Z9DIS2)
LNKLST ADD NAME(LNKLST00) DSN(SYS1.SHASLNKE) VOLUME(&SYSR1)
LNKLST ADD NAME(LNKLST00) DSN(SYS1.SIEAMIGE) VOLUME(&SYSR1)
LNKLST ACTIVATE NAME(LNKLST00)
```
Inspecting System Libraries

- If you want to find a member of the PARMLIB or PROCLIB concatenations, you can use the SYSLIB command on Marist
- The command builds lists of all members in the concatenation
- The individual members are listed under Data Set Name
- Use the “S” elect action to search
# SYSLIB

<table>
<thead>
<tr>
<th>Volume</th>
<th>Disposition</th>
<th>Act</th>
<th>DDname</th>
<th>Data Set Name</th>
<th>Actions: B E V M F C I Q</th>
</tr>
</thead>
<tbody>
<tr>
<td>ZOSYS1</td>
<td>SHR,KEEP</td>
<td></td>
<td>$PARMLIB</td>
<td>SYS1.MARIST.PARMLIB</td>
<td></td>
</tr>
<tr>
<td>Z0RES1</td>
<td>SHR,KEEP</td>
<td></td>
<td></td>
<td>SYS1.PARMLIB</td>
<td></td>
</tr>
<tr>
<td>MAR16</td>
<td>SHR,KEEP</td>
<td></td>
<td>$PROCLIB</td>
<td>SYS1.MARIST.PROCLIB</td>
<td></td>
</tr>
<tr>
<td>Z0RES1</td>
<td>SHR,KEEP</td>
<td></td>
<td></td>
<td>ADCD.219.PROCLIB</td>
<td></td>
</tr>
<tr>
<td>Z0RES3</td>
<td>SHR,KEEP</td>
<td></td>
<td></td>
<td>CEE.SCEPROC</td>
<td></td>
</tr>
<tr>
<td>Z0RES2</td>
<td>SHR,KEEP</td>
<td></td>
<td></td>
<td>CSG600.SCSQPDC</td>
<td></td>
</tr>
<tr>
<td>Z0RES2</td>
<td>SHR,KEEP</td>
<td></td>
<td></td>
<td>EUL.SEUFPDC</td>
<td></td>
</tr>
<tr>
<td>Z0RES2</td>
<td>SHR,KEEP</td>
<td></td>
<td></td>
<td>ICE.SICEPROC</td>
<td></td>
</tr>
<tr>
<td>Z0RES2</td>
<td>SHR,KEEP</td>
<td></td>
<td></td>
<td>EDY.SEDYPROC</td>
<td></td>
</tr>
<tr>
<td>Z0RES2</td>
<td>SHR,KEEP</td>
<td></td>
<td></td>
<td>HLA.SASHSAM1</td>
<td></td>
</tr>
<tr>
<td>Z0RES3</td>
<td>SHR,KEEP</td>
<td></td>
<td></td>
<td>CBC.SCOFPC</td>
<td></td>
</tr>
<tr>
<td>Z0RES1</td>
<td>SHR,KEEP</td>
<td></td>
<td>$TCPARM</td>
<td>SYS1.TCPPARMS</td>
<td></td>
</tr>
<tr>
<td>Z0RES1</td>
<td>SHR,KEEP</td>
<td></td>
<td>$VTAMLST</td>
<td>SYS1.MARIST.VTAMLST</td>
<td></td>
</tr>
<tr>
<td>Z0/01</td>
<td>SHR,KEEP</td>
<td></td>
<td></td>
<td>USER.VTAMLST</td>
<td></td>
</tr>
<tr>
<td>Z0RES1</td>
<td>SHR,KEEP</td>
<td></td>
<td></td>
<td>ADCD.219.VTAMLST</td>
<td></td>
</tr>
</tbody>
</table>

**Command === Scroll === PAGE**

F1=Help  F2=Split  F3=Exit  F5=RTind  F7=Up  F8=Down  F9=Swap

F16=Left  F11=Right  F12=Cancel
Concatenations

- PARMLIB concatenation is searched for z/OS system configuration parameters.
- The job procedure library (or PROCLIB concatenation) is searched for JCL PROCs.
- LPALST and LINKLIB concatenations are searched when a program is requested.

\[2\] A modern alternative is the use of JCLLIB.
Loading Programs

- Programs ("load modules") must be in central storage and therefore in the virtual storage of the address space before they can run.

- z/OS has a defined search order for a newly requested program.
Search Order for Programs

- STEPLIB (if present)
- JOBLIB (if present and no STEPLIB)
- Link Pack Area (LPA) concatenation:
  - Dynamic LPA modules
  - Fixed LPA
  - Modified LPA
  - Pageable LPA

- Linklist concatenation

---

\[ ^3 \text{See SYS1.MARIST.PARMLIB(LPALST8W)} \]
Defining the Link Pack Area

- **LPA** is built at IPL time from modules defined in LPALSTxx member of PARMLIB.
- **SYS1.LPALIB** is always the first library used (unless overridden by a SYSLIB statement).
- **LPA modules** are loaded into common storage at IPL time and available to all address spaces.
Example LPALSTxx file

SYS2.LPALIB,
SYS1.LPALIB,
SYS1.SERBLPA,
SDF2.V1R4M0.SDGILPA,
SYS1.SIATLPA,
IGN.INGM0D3,
NETVIEW.SCNMPA1,
REXX.V1R3M0.SEGALPA,
LPA Types

- Fixed LPA are those modules defined in IEAFLIXxx (fixed in central storage)
- Pagable LPA are most other modules (pages eligible to be stolen)
- Modified LPA has modules that are temporary replacements for PLPA modules (used for testing updates for a particular IPL)
Example IEAFIXxx file

INCLUDE LIBRARY(SYS1.LPALIB)
MODULES(IEAVAR00,
IEAVAR06,
IGC001G,
ICHRFC00,
ICHRFR00)
INCLUDE LIBRARY (SYS1.SVCLIB)
MODULES(IGC09302)
System Symbols

- System symbols allow the use of a shared PARMLIB by two or more systems
- Each symbol has a name which can be used in various places and then substituted at IPL time
- Major uses are indirect cataloging and substituting system specific datasets such as the page data sets
System Symbols

- Static System Symbols are defined at IPL and remain fixed for the life of an IPL
- Dynamic System Symbols can change at any point
Defining System Symbols

System symbols are defined in a IEASYSMxx PARMLIB member:

SYSDEF   HWNAME(SCZP801)
LPARNAME(A08)
SYSNAME(SC04)
SYSPARM(R3,04)
SYMDEF(&CPCNAME='P801')
SYMDEF(&DFHSMHST='ON')
SYMDEF(&SYSR2='ZXYSY2')
SYMDEF(&SYSR3='&SYSR1(1:5).3')
Using System Symbols

To use a system symbol, use “&SYMNAME.” where the substitution is needed:

PAGE=(PAGE.&SYSNAME..&PLPADSN1.,
    PAGE.&SYSNAME..&COMMDSN1.,
    PAGE.&SYSNAME..LOCAL1,
    PAGE.&SYSNAME..LOCAL2,L)
IPLing z/OS

• IPL = Initial Program Load = “booting”

• Specify address of the IPL volume and the IODF volume

• Bootstrap code on IPL volume is loaded into storage (at address 0) and control is passed to it

• Bootstrap reads the IPLTEXT program IEAIPL00 which is given control
IEAIIPL00

- Clears central storage areas (to zeros), defines storage areas for master scheduler
- Locates SYS1.NUCLEUS, loads modules to construct environment of control blocks and subsystems
- Finally, the Nucleus Initialization Program (NIP) is loaded
Nucleus Initialization Program

NIP sets up:

- System Queue Area (SQA)
- PLPA, FLPA, MLPA
- Common Service Area (CSA)
- Starts the master scheduler
The System Queue Area (SQA)

- Storage that is common to all running tasks
- Contains tables and queues with system meta data
- Contents are highly dependent on configuration
Master Scheduler (ASID = 1)

- Starts other required address spaces
- Initializes subsystems, including JES
- Once JES is ready, jobs can be started
- VTAM and TSO started after JES

⇒ System available to do work!
Virtual Input/Output (VIO)

- Method of using memory to store small temporary data sets for rapid access
- These are backed up to disk (unlike a RAM disk on a PC)
- Size is restricted

Configured on Marist in SYS1.MARIST.PARMLIB(IEASYS99) to be SYS1.STGINDEX.
IPL Types

- **Cold start:** PLPA is reloaded and VIO is cleared (required when contents of LPA have changed)

- **Quick start:** PLPA is not reloaded, but VIO is cleared (no changes to LPA made, but VIO needs to be refreshed)

- **Warm start:** PLPA and VIO are retained from before the IPL, jobs can restart using journalled VIO data
Libraries required at IPL time

These must be present for the system to IPL:

- SYS1.PARMLIB
- SYS1.IPLPARM – with LOADxx member!
- SYS1.LPALIB
- SYS1.PROCLIB
- SYS1.NUCLEUS
Specifying IPL Information

- Operator specifies IODF file's device address, selects LOADxx member and controls how the system will prompt during IPL.

- If IEASYSxx is not specified in LOADxx, then the operator will be prompted to specify system parameters.
LOAD\textsubscript{xx}

The LOAD\textsubscript{xx} member specifies:

- The IODF data set name
- The master catalog name and volume
- The parmlib concatenation
- The IEASY\textsubscript{xx} member (of SYS1.PARMLIB) to use ($xx = 00$ is default)
- The IEASYM\textsubscript{xx} member (of SYS1.PARMLIB) to use ($xx = 00$ is default)
Locating LOADxx

LOADPARM defines the address of the IODF and the suffix of the LOADxx member. The system searches for LOADxx in:

- SYS1.IPLPARM on IODF volume
- SYS1.PARMLIB on IODF volume
- SYS1.PARMLIB on the IPL volume
Example LOADxx file

IODF   00 SYS6 MOEMVSP1 01 Y
SYSCAT MPAT1113CATALOG.MCAT.VMPCAT1
HWNAME P201
LPARNAME A1
PARMLIB SYS0.IPLPARM
PARMLIB SYS1.OS390R7.PARMLIB
PARMLIB SYSPROG.SYS1.PARMLIB
Search order for Control Parameters

Using the LOADxx file from the previous slide:

1. SYS0.IPLPARM
2. SYS1.OS390R7.PARMLIB
3. SYSPROG.SYS1.PARMLIB
4. **SYS1.PARMLIB** – implicitly last!

LOADxx can specify up to 16 PDS names.
IPL at Marist

- Master configuration for the IPL is in SYS1.IPLPARM(LOAD99)
- This specifies the PARMLIB concatenation and other major parameters
- If IEASYSYxx is not specified in LOADxx, then the operator will be prompted to specify system parameters!
SDSF /D IPLINFO

RESPONSE=SOW1

IEE254I 17.42.47 IPLINFO DISPLAY 283
SYSTEM IPLED AT 10.09.45 ON 11/09/2008
RELEASE z/OS 01.08.00 LICENSE = z/OS
USED LOADW1 IN SYS1.IPLPARM ON OCE3
ARCHLVL = 2 MTLSHARE = N
IEASYM LIST = (W1,SV,VN)
IEASYS LIST = 18 (OP)
IODF DEVICE OCE3
IPL DEVICE 1000 VOLUME VIMVSB
SDSF /D PARMLIB

RESPONSE=S0W1

IEE251I 17.44.27 PARMLIB DISPLAY 291
PARMLIB DATA SETS SPECIFIED
AT IPL

<table>
<thead>
<tr>
<th>ENTRY</th>
<th>FLAGS</th>
<th>VOLUME</th>
<th>DATA SET</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>S</td>
<td>VPMVSD</td>
<td>VENDOR.PARMLIB</td>
</tr>
<tr>
<td>2</td>
<td>S</td>
<td>VTMVSG</td>
<td>SVTSC.PARMLIB</td>
</tr>
<tr>
<td>3</td>
<td>S</td>
<td>VTLVLO</td>
<td>LVLO.PARMLIB</td>
</tr>
<tr>
<td>4</td>
<td>D</td>
<td>VIMVSB</td>
<td>SYS1.PARMLIB</td>
</tr>
</tbody>
</table>
IPL Errors

• If a required PARMLIB member has an error (or is missing), the operator will be prompted.

• If the parameter cannot be corrected, then the default can be accepted.

• If a default does not exist, then the parameter maybe cancelled or the IPL abandoned.
Shutting down the System

- Production systems are only shutdown when necessary (change, work on power lines)
- Many changes can be done dynamically without IPL
- Tasks need to be shut down in the correct order (installation specific)
- Most installations have an automation package for shutdown
Questions

?