Open Source on the Modern Mainframe

Elizabeth K. Joseph
@pleia2
Hello! My name is Lyz.

My background is as a Linux Systems Administrator, a role I held for almost 15 years.

I've been involved with open source projects for nearly 20 years!

Open source is the common thread in my systems and operations career.

I'm speaking as a community member interested in open source on the mainframe :)

This is a lightning talk, we'll go fast!

Slides will be available on https://princessleia.com (maybe they're up already!)

Questions? Comments? Please reach out! lyz@princessleia.com

Tweet at me! @pleia2
What is a mainframe?
What is a mainframe?

Not x86.
(it's s390x!)

190 5.2ghz processor units (PUs), with 12 cores per chip

But also...

- 40TB of RAM
- 60 PCIe control units across 12 PCIe I/O drawers
- 22 dedicated I/O offload processors (SAPs) pre-allocated per system
What do mainframes run?

- Linux! (SLES, RHEL, Ubuntu...)
- A ton of open source software!
- Several proprietary mainframe operating systems (but we're here to talk about open source!)
In 1955, the volunteer-run SHARE Inc was founded.

A key resource for this organization was the SHARE library of software that systems programmers would share among their peers, freely.

In 1959, SHARE released the SHARE Operating System (SOS), one of the first true "operating systems"¹ and Wikipedia says of SOS:

"SOS was one of the first instances of "commons-based peer production" now widely used in the development of free and open-source software such as Linux and the GNU project."

¹ https://en.wikipedia.org/wiki/SHARE_(computing)
History of Linux on the mainframe

Started out as the "Bigfoot" (i370) port by several community members in 1998-99.

IBM released the first Linux kernel patches to support s390x in December 1999.

In October 2000, SUSE Linux Enterprise Server became the first, still in production, enterprise Linux to support s390x.

Red Hat quickly followed as the second, still in production, enterprise Linux for the mainframe.

Ubuntu support was announced in 2016 and began with Ubuntu 16.04.
Open Source for Linux
Open Source Hardware support on Linux

You may be tempted to believe there's proprietary "sauce" for the hardware encryption support on Linux on the mainframe, but it's just the standard open source libraries and tools we know and love!

- dm-crypt
- OpenSSL and libcrypto (including for ssh, scp, sftp, Apache mod_ssl...)
- IPSec
- libica crypto library for s390x (https://github.com/opencryptoki/libica)
- aes for Go (https://golang.org/pkg/crypto/aes/)
Open Source Docker Images
Open Mainframe Project

- Full project hosting, including code, and mailing lists
- Blogs and podcasts of general interest to the open source mainframe community
- Slack and forums for communication among participants
- Project support for 3rd party open source projects seeking infrastructure (VMs, CI/CD services)
- New in 2020: Annual conference!
Software Discovery Tool

• Just launched in 2020!
• First release will search SLES, RHEL, and Ubuntu for open source packages built for the mainframe
• Learn more and contribute at https://github.com/openmainframeproject/software-discovery-tool
At the core, Zowe provides an API that allows easier, unified access to the mainframe.

This has allowed an ecosystem of tools to flourish.

- A CLI that can be installed on any machine for interacting with the mainframe
- "Desktop" style client that allows graphical access to mainframe data
- DevOps and related CI/CD tooling
- VS Code integration for developers
Your open source software on the mainframe

Free Tooling from the distributions

Ubuntu

Ubuntu Personal Package archives on Launchpad.net [https://help.launchpad.net/Packaging/PPA](https://help.launchpad.net/Packaging/PPA)

SUSE

OpenSUSE build service [https://build.opensuse.org/](https://build.opensuse.org/)
Your open source software on the mainframe

Open Mainframe Project Supported Projects Virtual Machines, hosted by Marist College

https://www.openmainframeproject.org/projects/supported-projects
Your open source software on the mainframe

Free CI/CD Tooling

Jenkins instance for s390x maintained by the Oregon State University Open Source Lab (OSU OSL): [s390x Development Hosting](#)

TravisCI build service for s390x (Beta trial for open source projects): [Building on Multiple CPU Architectures](#)
But what language?

Source code across architectures will generally be identical, but it needs to be compiled (C, C++), or otherwise interpreted (Python, Node.js) for this architecture.

That means you need a compiler or an interpreter built for the mainframe.
Language tips

The mainframe architecture is big-endian, but most of the supported architectures today are little-endian.

Your code will probably build and run, give it a try!

If not, the higher level a language is, the more luck you'll have. Lower level languages like C do more hardware-specific operations, higher level languages like Node.js have much of that abstracted away.
Thank you!