



COMPILING SOFTWARE & USING LINUX ON IBM Z

ELIZABETH K. JOSEPH, IBM

MOM RELAUNCH, MAY 2021

COMPILING SOFTWARE 101 – ARCHITECTURE

IBM Z uses the s390x hardware architecture, most servers today use x86, with others using ARM or Power.

Most developers don't pay much attention to architecture! But there will be an increasing need to do so as non-x86 architectures become more common.

COMPILING SOFTWARE 101 – CODE

- At the lowest levels, classic* computing still only understands 0 and 1. That's what all those billions of tiny transistors are doing.
- Compilers and interpreters take human-readable code that you write and convert it to something the computer can understand, ultimately a series of 0s and 1s.
- The code you see, is just the first step in the process as far as the computer is concerned.

* What is beyond Classic Computing? Quantum!

COMPILING SOFTWARE 101 – OPEN SOURCE

When something is "open source" you have access to the human-readable code, it's available in the open.



You then compile that code to create a binary. This code must be compiled for the respective architecture you're targeting since it needs to be built for that CPU hardware (x86, s390x, ARM, Power, etc).



TODAY WE'RE GOING TO COMPILE
SOME SOFTWARE FOR IBM Z!

LINUXONE COMMUNITY CLOUD

If you haven't already, get a free LinuxONE Community Cloud Virtual Machine: <https://linuxone.cloud.marist.edu/#/register?flag=VM>

And learn more at <https://developer.ibm.com/linuxone>

Use the event code: LAUNCHTOZ21

Follow the Virtual Server Deployment Guide: <https://github.com/linuxone-community-cloud/technical-resources/blob/master/faststart/deploy-virtual-server.md> and launch a RHEL8.3 server.

Tip: If you've never used SSH before, logging into the server may be the trickiest part of this whole tutorial! Let us know if you need help.

LOG IN AND EXPLORE!

```
$ cd ~
```

```
$ mkdir my_directory
```

```
$ cd my_directory
```

```
$ touch a_file
```

```
$ touch another_file
```

```
$ ls
```

```
a_file
```

```
another_file
```

```
$ rm a_file
```

```
$ cd /
```

```
$ ls -l
```

bin - where programs (binaries) are kept

etc - System-wide configuration files

home - Where users keep their own files, including you!

lib - Short for “library”. Shared code and kernel libraries

root - Where the system administrator (root) keeps their files

usr - Shared, read-only files, which includes some binaries and documentation

var - Short for “variable” files. Items in here change frequently, including logs, printer spools, and some caches.

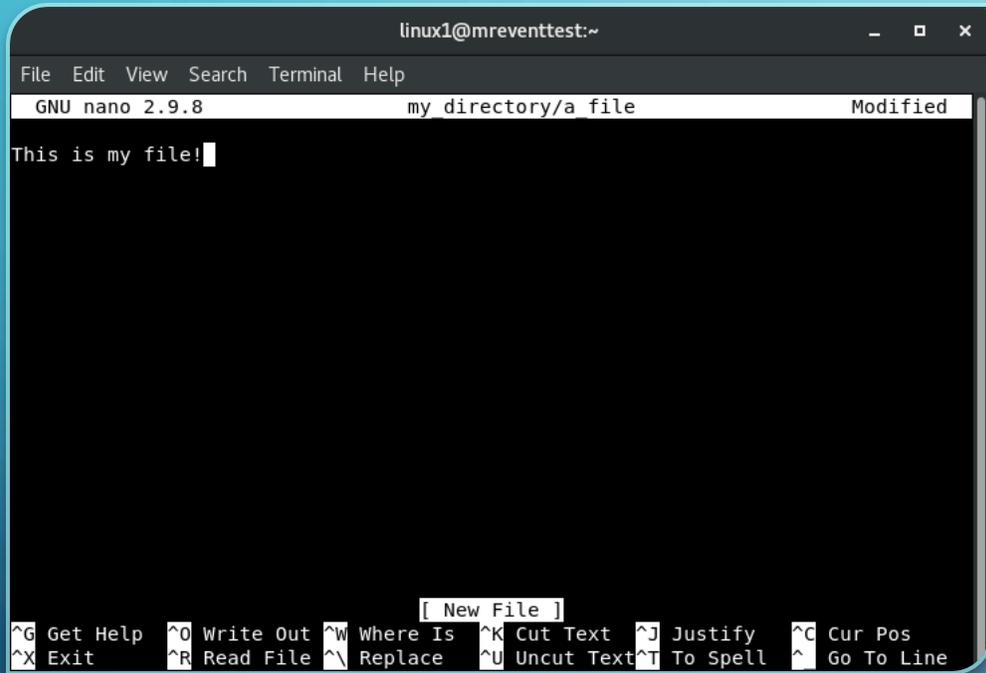
IS THIS REALLY IBM Z? PEEK AT THE HARDWARE

```
$ lscpu
```

```
$ cat /proc/cpuinfo
```

```
linux1@mreventtest:~  
File Edit View Search Terminal Help  
[linux1@mreventtest ~]$ lscpu  
Architecture:          s390x  
CPU op-mode(s):        32-bit, 64-bit  
Byte Order:            Big Endian  
CPU(s):                2  
On-line CPU(s) list:  0,1  
Thread(s) per core:    1  
Core(s) per socket:    1  
Socket(s) per book:    1  
Book(s) per drawer:    1  
Drawer(s):             2  
NUMA node(s):          1  
Vendor ID:             IBM/S390  
Machine type:          8561  
CPU dynamic MHz:       5200  
CPU static MHz:        5200  
BogoMIPS:              3241.00  
Hypervisor:            z/VM 7.1.0  
Hypervisor vendor:     IBM  
Virtualization type:   full  
Dispatching mode:      horizontal  
L1d cache:             128K  
L1i cache:             128K  
L2d cache:             4096K
```

EDITING AND VIEWING FILES



```
linux1@mreventtest:~  
File Edit View Search Terminal Help  
GNU nano 2.9.8 my directory/a file Modified  
This is my file!  
[ New File ]  
^G Get Help ^O Write Out ^W Where Is ^K Cut Text ^J Justify ^C Cur Pos  
^X Exit ^R Read File ^\ Replace ^U Uncut Text ^T To Spell ^_ Go To Line
```

```
$ cd
```

```
$ nano my_directory/a_file
```

(ctrl x to exit, it will ask to save and confirm file name)

```
$ sudo tail -f /var/log/messages
```

LET'S BUILD A WEBSITE!

```
$ sudo yum install httpd
$ sudo systemctl start httpd
$ sudo systemctl enable httpd
$ sudo systemctl status httpd
$ sudo nano /var/www/html/index.html
```

Write some HTML! Or use this:

```
<html>
```

```
<body>
```

```
<h1>This is my website!</h1>
```

```
</body>
```

```
</html>
```

LET THE TRAFFIC IN

```
$ sudo iptables -I INPUT -p tcp --dport 80 -j ACCEPT  
$ sudo iptables -S
```

Navigate to the IP address of your server in your web browser to see your site!

COMPILING SOFTWARE – DOWNLOAD & EXTRACT

Install pre-reqs:

```
$ sudo yum install make gcc
```

(respond with "y" to accept installing everything)

Download the source code:

```
$ curl -LO https://github.com/vicgeralds/vitetris/archive/v0.58.0.tar.gz
```

Extract the code:

```
$ tar -xvf v0.58.0.tar.gz
```

COMPILING SOFTWARE — CONFIGURE AND MAKE

Now that you have a directory for the vitetris software, navigate into that directory and run the configure command to confirm you have everything you need to build it (you should):

```
$ cd vitetris-0.58.0/
```

```
$ ./configure
```

And then run the make command to compile your software:

```
$ make
```

Score
000221

Top Scores
1. 023298

COMPILING SOFTWARE — PLAY!

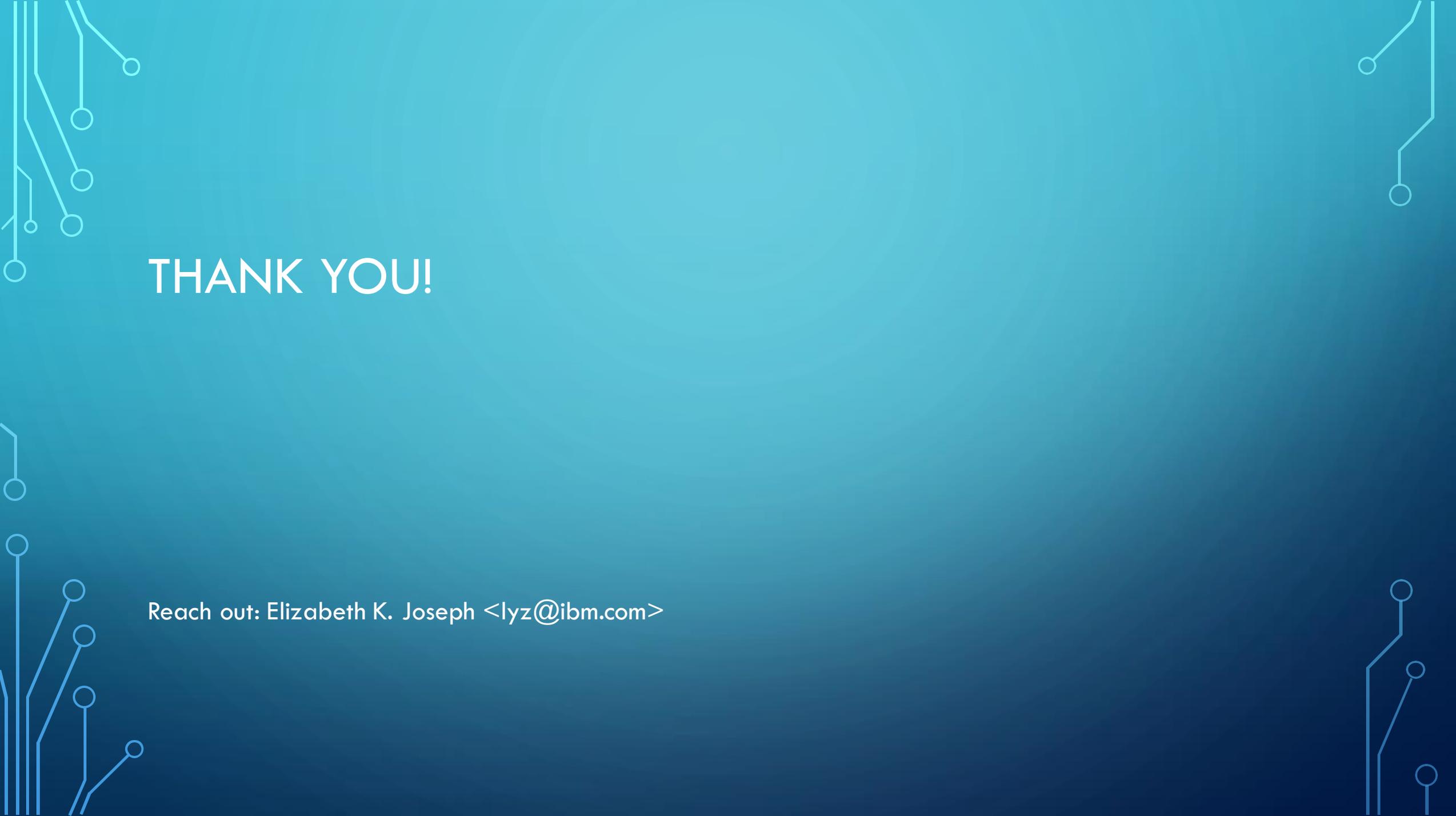
Level
000

Lines
000

```
$ ./tetris
```

#I	002
#J	007
#L	003
#O	001
#S	001
#T	005
#Z	003

Sum 0022

The background is a solid teal color. In the corners, there are decorative white line-art patterns resembling circuit traces or data paths, with small circles at the end of the lines.

THANK YOU!

Reach out: Elizabeth K. Joseph <lyz@ibm.com>