COMPILING SOFTWARE & USING LINUX ON IBM Z

ELIZABETH K. JOSEPH, IBM

MOM RELAUNCH, MAY 2021
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.
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!
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!
If you haven't already, get a free LinuxONE Community Cloud Virtual Machine: [https://linuxone.cloud.marist.edu/#/register?flag=VM](https://linuxone.cloud.marist.edu/#/register?flag=VM)

And learn more at [https://developer.ibm.com/linuxone](https://developer.ibm.com/linuxone)

Use the event code: LAUNCHTOZ21


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

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.
ADMINISTRATOR ACCESS & INSTALLING SOFTWARE

$ ls /root
ls: cannot open directory '/root': permission denied

$ sudo ls /root

$ sudo yum install nano
IS THIS REALLY IBM Z? PEEK AT THE HARDWARE

$ lscpu
$ cat /proc/cpuinfo
EDITING AND VIEWING FILES

$ cd

$ nano my_directory/a_file

(ctl 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!
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
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
```
COMPILING SOFTWARE – PLAY!

$ ./tetris
THANK YOU!

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