

Linux on Z: Go Worldwide

Take control of your Linux instance and start building

 15 steps  90 minutes

THE CHALLENGE

We're going to take a slight break from our z/OS-based challenges for a while and focus on another important aspect of enterprise IT. Linux started as a hobbyist alternative to expensive UNIX systems, which were only available to those in universities or large companies. Through rapid adoption, it landed on servers, laptops, phones, and devices all around the world. Of course, one of those servers is the IBM Z mainframe, where it benefits from the extreme resiliency, performance, and scalability that the platform offers.

BEFORE YOU BEGIN

Access to a Linux system is required for this challenge, so you'll definitely want to complete the first Linux challenge, LNX1, before attempting these steps.

```
httpd-filesystem.noarch : The basic directory layout for the Apache HTTP server
python38-mod_wsgi.s390x : A WSGI interface for Python web applications in Apache
mod_ldap.s390x : LDAP authentication modules for the Apache HTTP Server
python3-mod_wsgi.s390x : A WSGI interface for Python web applications in Apache
mod_proxy_html.s390x : HTML and XML content filters for the Apache HTTP Server
mod_auth_mellon.s390x : A SAML 2.0 authentication module for the Apache Httpd Server
mod_md.s390x : Certificate provisioning using ACME for the Apache HTTP Server
pcp-export-pcp2spark.s390x : Performance Co-Pilot tools for exporting PCP metrics to Apache Spark
mod_intercept_form_submit.s390x : Apache module to intercept login form submission and run PAM authentication
mod_lookup_identity.s390x : Apache module to retrieve additional information about the authenticated user
MTM> sudo yum install httpd.s390x
Last metadata expiration check: 0:06:28 ago on Mon 17 Aug 2020 09:30:18 AM EDT.
Dependencies resolved.
=====
Transaction Summary
=====
Install 10 Packages
=====
Total download size: 1.9 M
Installed size: 5.6 M
Is this ok [y/N]: y
```

```
MTM> sudo apachectl start
MTM> sudo chkconfig httpd on
Note: Forwarding request to 'systemctl enable httpd.service'.
Created symlink /etc/systemd/system/multi-user.target.wants/httpd.service → /usr/lib/systemd/sys-
.service.
MTM> sudo apachectl status
● httpd.service - The Apache HTTP Server
   Loaded: loaded (/usr/lib/systemd/system/httpd.service; enabled; vendor preset: disabled)
     Active: active (running) since Mon 2020-08-17 09:43:51 EDT; 30s ago
       Docs: man:httpd.service(8)
   Main PID: 21345 (httpd)
      Status: "Running, listening on: port 80"
         Tasks: 213 (limit: 24563)
        Memory: 21.6M
      CGroup: /system.slice/httpd.service
              ├─21345 /usr/sbin/httpd -DFOREGROUND
              ├─21346 /usr/sbin/httpd -DFOREGROUND
              ├─21347 /usr/sbin/httpd -DFOREGROUND
              ├─21348 /usr/sbin/httpd -DFOREGROUND
              └─21349 /usr/sbin/httpd -DFOREGROUND

Aug 17 09:43:51 ibm-jeff-bisti.novalocal systemd[1]: Starting The Apache HTTP Server...
Aug 17 09:43:51 ibm-jeff-bisti.novalocal systemd[1]: Started The Apache HTTP Server.
Aug 17 09:43:57 ibm-jeff-bisti.novalocal httpd[21345]: Server configured, listening on: port 80
MTM>
```

GNU nano 2.9.8 index.html

```
<!DOCTYPE html>
<html>
<body>
<h1>This is my web page</h1>
<p>... and it just happens to be running on IBM Z hardware!</p>
</body>
</html>
```

**^G Get Help ^O Write Out ^W Where Is ^K Cut Text ^J Justify
^X Exit ^R Read File ^\ Replace ^U Uncut Text ^T To Spell**

1. INSTALL A WEB SERVER

Type **yum search apache** to look for installable packages related to the [Apache web server](#).

The first result, **httpd.s390x** is the one we want. As a note, s390x is shorthand for 64 bit Z architecture. Install that package with **sudo yum install httpd.s390x**

It will show you a number of dependencies that it also needs to install, and ask you if this is ok. Enter 'y' to tell it to proceed.

2. START AND ENABLE APACHE

We've installed the Apache web server, but it won't do anything until we start it. Let's start it, enable it so it'll start automatically, and then check on its status.

sudo apachectl start
sudo chkconfig httpd on
sudo apachectl status

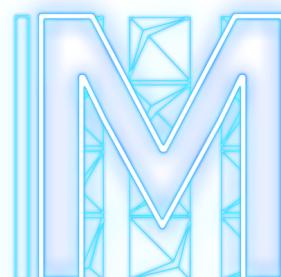
If the output of the "status" command is longer than can fit in your window, you may need to scroll to the bottom, or press Q to quit out.

3. BUILD A WEB PAGE

Go into **/var/www/html** and create a file called **index.html** (*Remember that you'll need root authority to write out here...*)

If you're familiar with HTML, then go nuts. Otherwise, feel free to copy/paste the sample code above. When you're done, **Control-O** to save and **Control-X** to quit nano.

I know this doesn't need to be said, but don't put anything out here in poor taste or questionable judgement. Actions will be taken for improper use of this system.



```
[linux1@mtmfwtest ~]$ ifconfig
enc1000: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    inet 148.100.76.94 netmask 255.255.252.0 broadcast 148.100.79.255
        inet6 fe80::a1:41ff:fe2:4351 prefixlen 64 scopeid 0x20<link>
    ether 02:a1:41:c2:43:51 txqueuelen 1000 (Ethernet)
    RX packets 1271 bytes 50640 (49.4 KiB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 196 bytes 71981 (70.2 KiB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING> mtu 65536
    inet 127.0.0.1 netmask 255.0.0.0
    inet6 ::1 prefixlen 128 scopeid 0x10<host>
    loop txqueuelen 1000 (Local Loopback)
    RX packets 0 bytes 0 (0.0 B)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 0 bytes 0 (0.0 B)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

[linux1@mtmfwtest ~]$
```

4. FIND YOUR IP ADDRESS

Before we go much further, make sure you have your system's IP address handy. It's listed on the Community Cloud page where you created this server instance, or you can issue an **ifconfig** (interface config) command and look for the inet address on the enc1000 interface (highlighted in the screenshot above)

Write that down or just make a note of where you can find it.

WHY DO I HAVE TO KEEP TYPING SUDO?

Maybe you've heard the expression "Measure twice, cut once". The idea there being that if you rush during the planning stages, you're likely to make a mistake. If only you had taken another few seconds to think about what you were about to do, it could have saved a lot of time and heartache.

By using a default non-privileged user, and only elevating your access when you *really* need it, you're less likely to accidentally erase an important file or directory. It reminds you that you're about to do something that could have potentially dangerous consequences.

Whenever you enter a command after sudo, take a few seconds to double-check your command.

```
MTM> sudo service firewalld start
Redirecting to /bin/systemctl start firewalld.service
MTM> sudo firewall-cmd --zone=public --add-port=80/tcp --permanent
success
MTM> sudo firewall-cmd --reload
success
MTM> sudo systemctl enable firewalld
Created symlink /etc/systemd/system/dbus-org.fedoraproject.FirewallD1.service →
/usr/lib/systemd/system/firewalld.service.
Created symlink /etc/systemd/system/multi-user.target.wants/firewalld.service →
/usr/lib/systemd/system/firewalld.service.
MTM>
```

5. YOU'RE MY FIREWALL

By default, the Community Cloud blocks all incoming access other than SSH. This is a security precaution. To look at our web page, we need to make an exception in the firewall for port 80, which the web server is running on. Use these commands:

```
sudo service firewalld start
sudo firewall-cmd --zone=public --add-port=80/tcp --permanent
sudo firewall-cmd --reload
sudo systemctl enable firewalld
```

```
MTM> sudo reboot
Connection to 148.100.76.94 closed by remote host.
Connection to 148.100.76.94 closed.
Monorail:~ jbisti$ ssh -i MTMFTEST.pem linux1@148.100.76.94
[MTMFTEST:~] Last login: Wed Aug 26 15:07:03 2020 from 67.251.217.36
[linux1@mtmfwtest ~]$ sudo firewall-cmd --reload
success
[linux1@mtmfwtest ~]$
```

Welcome to the IBM LinuxONE Community Cloud!
This server is for authorized users only. All activity is logged and monitored.
Individuals using this server must abide to the Terms and Conditions listed here:
<https://www.ibm.com/community/z/ibm-linuxone-community-cloud-terms-and-conditions/>
Your access will be revoked for any non-compliance.

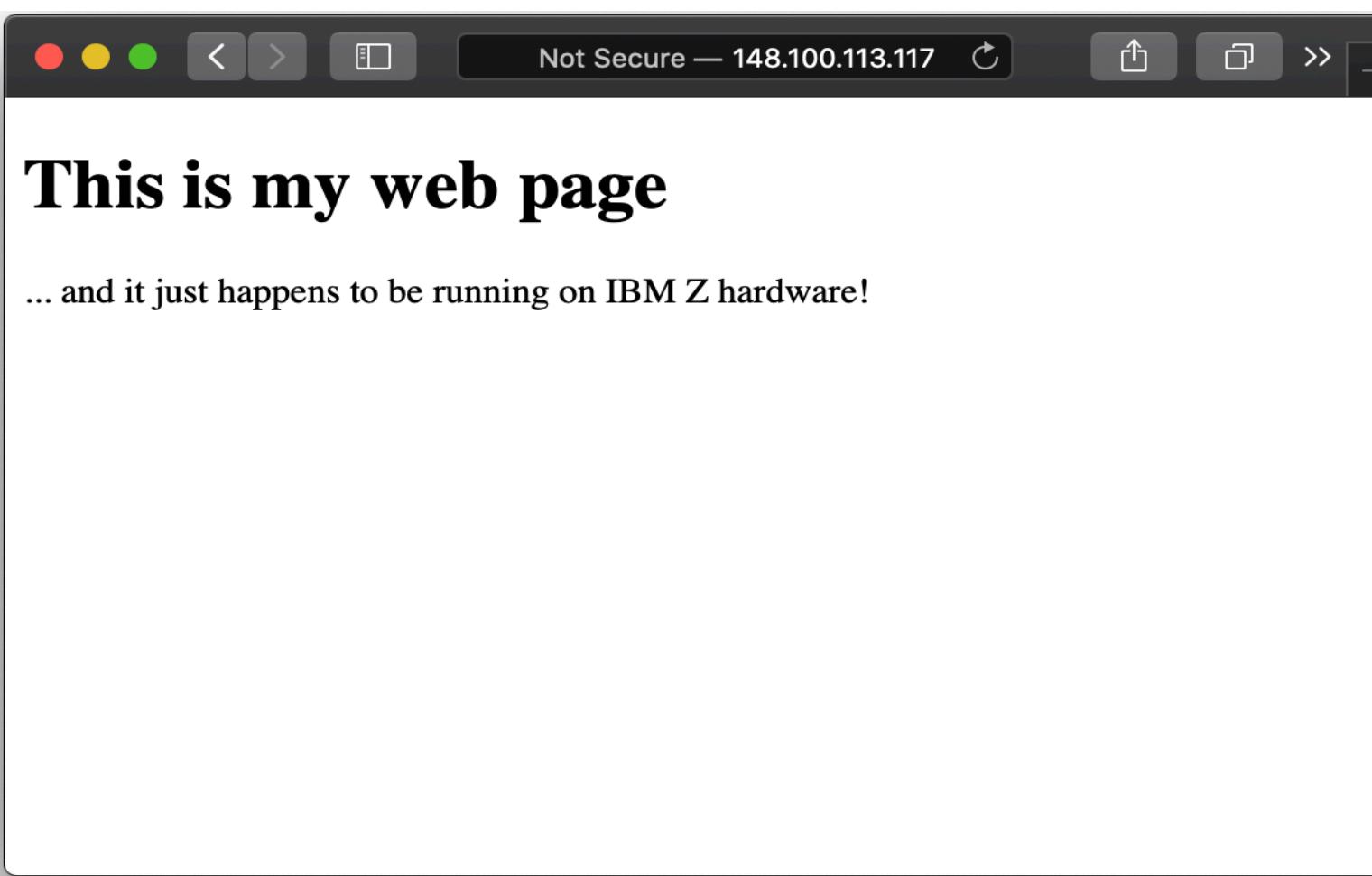
6. IF YOUR SYSTEM REBOOTS

You may need to reload the firewall changes if your system goes down, either because you rebooted it, or it was taken down for maintenance. This step is here *just in case*. In other words don't reboot your system unless you really really want to. Uptime is a beautiful thing.

Changes made to the firewall won't stick after a reboot or outage, and you'll need to SSH back in to issue the command **sudo firewall-cmd --reload** again.

After that, you should be able to access your running webserver on port 80.





7. VIEW YOUR MASTERPIECE

Open a web browser and open go the the IP address of your system. For example, <http://148.100.244.250>

If the server is up, the firewall is modified, and you edited the HTML, you should be able to pull up your creation in a web browser.

WHY DO I HAVE TO KEEP TYPING SUDO?

Maybe you've heard the expression "Measure twice, cut once". The idea there being that if you rush during the planning stages, you're likely to make a mistake. If only you had taken another few seconds to think about what you were about to do, it could have saved a lot of time and heartache.

By using a default non-privileged user, and only elevating your access when you *really* need it, you're less likely to accidentally erase an important file or directory. It reminds you that you're about to do something that could have potentially dangerous consequences.

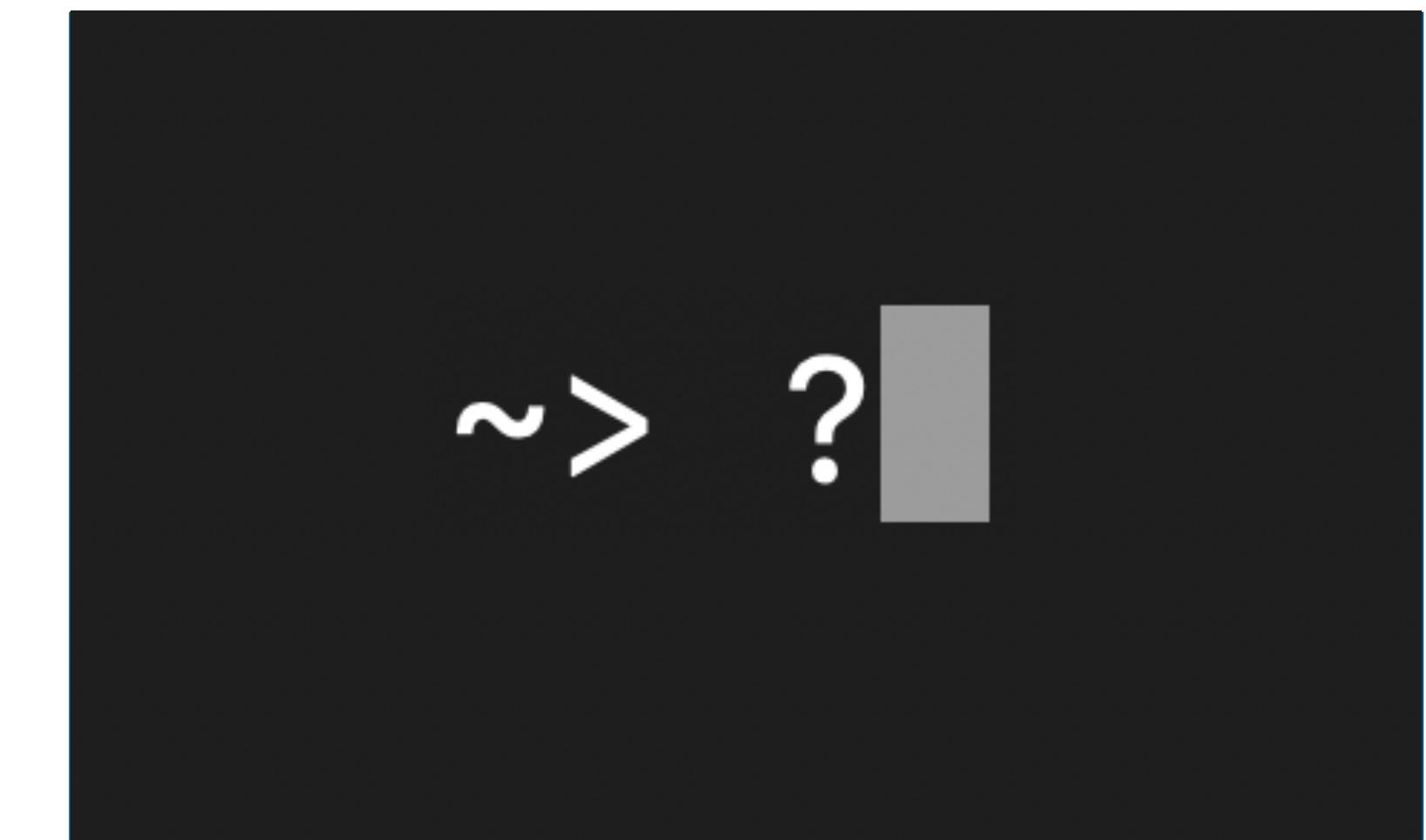
Whenever you enter a command after sudo, take a few seconds to double-check your command.

```
MTM> sudo tail -f /var/log/httpd/access_log
17.255.255.255 - - [17/Aug/2020:10:33:13 -0400] "GET / HTTP/1.1" 304 -
Intel Mac OS X 10_15_6) AppleWebKit/605.1.15 (KHTML, like Gecko) Version
17.255.255.255 - - [17/Aug/2020:10:33:15 -0400] "GET / HTTP/1.1" 200 139
Intel Mac OS X 10_15_6) AppleWebKit/605.1.15 (KHTML, like Gecko) Version
17.255.255.255 - - [17/Aug/2020:10:33:16 -0400] "GET / HTTP/1.1" 200 139
Intel Mac OS X 10_15_6) AppleWebKit/605.1.15 (KHTML, like Gecko) Version
17.255.255.255 - - [17/Aug/2020:10:33:20 -0400] "GET / HTTP/1.1" 200 139
Intel Mac OS X 10_15_6) AppleWebKit/605.1.15 (KHTML, like Gecko) Version
17.255.255.255 - - [17/Aug/2020:10:33:20 -0400] "GET / HTTP/1.1" 200 139
Intel Mac OS X 10_15_6) AppleWebKit/605.1.15 (KHTML, like Gecko) Version
17.255.255.255 - - [17/Aug/2020:10:33:21 -0400] "GET / HTTP/1.1" 200 139
Intel Mac OS X 10_15_6) AppleWebKit/605.1.15 (KHTML, like Gecko) Version
17.255.255.255 - - [17/Aug/2020:10:33:22 -0400] "GET / HTTP/1.1" 200 139
Intel Mac OS X 10_15_6) AppleWebKit/605.1.15 (KHTML, like Gecko) Version
17.255.255.255 - - [17/Aug/2020:10:33:22 -0400] "GET / HTTP/1.1" 200 139
Intel Mac OS X 10_15_6) AppleWebKit/605.1.15 (KHTML, like Gecko) Version
17.255.255.255 - - [17/Aug/2020:10:33:22 -0400] "GET / HTTP/1.1" 200 139
Intel Mac OS X 10_15_6) AppleWebKit/605.1.15 (KHTML, like Gecko) Version
```

8. WATCH THE ACCESS

Enter the command `sudo tail -f /var/log/httpd/access_log`

This is a record of any time your web page has been requested, including the date/time requesting IP address, and browser information. Watch the access log move as you request your web page from a browser window. If you can, try to open it from another browser on another computer or phone and see the differences.



9. WOULD U LIKE 2 PLAY A GAME?

Sometimes the software you want isn't readily available through a manager like yum, or in the spirit of open source software, you want to make your own adjustments to the source code and build things on your own. That's all possible with the compilers available here.

In the next few steps, we'll download the source code for a game, Vitetris, extract the source code, and then compile that code so it'll run on your system.

If there's time left over at the end, I suppose you can do a little gaming, as a treat.

```
MTM> curl -LO https://github.com/vicgeralds/vitetrис/archive/v0.58.0.tar.gz
% Total    % Received % Xferd  Average Speed   Time   Time  Curr:
          Dload  Upload Total Spent   Left Speed
100 128 100 128    0     0 2666      0 --:--:-- --:--:-- 20
100 96957 0 96957    0     0 422k      0 --:--:-- --:--:-- 5
MTM> ls -lah
total 116K
drwx-----. 2 linux1 linux1 29 Jul 27 16:33 .ssh
-rw-rw-r--. 1 linux1 linux1 95K Aug 17 11:02 v0.58.0.tar.gz
MTM> tar xvf v0.58.0.tar.gz
vitetrис-0.58.0/
vitetrис-0.58.0/.gitignore
vitetrис-0.58.0/INSTALL
vitetrис-0.58.0/Makefile
vitetrис-0.58.0/README
vitetrис-0.58.0/changes.txt
vitetrис-0.58.0/config.mk
vitetrис-0.58.0/configure
vitetrис-0.58.0/icon.ico
vitetrис-0.58.0/icon.rc
vitetrис-0.58.0/licence.txt
vitetrис-0.58.0/pc8x16.fnt
vitetrис-0.58.0/src-conf.sh
vitetrис-0.58.0/src/
vitetrис-0.58.0/src/Makefile
vitetrис-0.58.0/src/cfgfile.c
vitetrис-0.58.0/src/cfgfile.h
MTM>
```

10. DOWNLOAD THE CODE

Download the game source code with the following command. Curl is a command-line utility for interacting with URLs. It supports FTP, HTTP, LDAP, and many other protocols. In this case, we're going to use it to download a file from a remote web server onto our Linux server.

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

```
drwx-----. 2 linux1 linux1 29 Jul 27 16:33 .ssh
-rw-rw-r--. 1 linux1 linux1 95K Aug 17 11:02 v0.58.0.tar.gz
MTM> tar xvf v0.58.0.tar.gz
vitetrис-0.58.0/
vitetrис-0.58.0/.gitignore
vitetrис-0.58.0/INSTALL
vitetrис-0.58.0/Makefile
vitetrис-0.58.0/README
vitetrис-0.58.0/changes.txt
vitetrис-0.58.0/config.mk
vitetrис-0.58.0/configure
vitetrис-0.58.0/icon.ico
vitetrис-0.58.0/icon.rc
vitetrис-0.58.0/licence.txt
vitetrис-0.58.0/pc8x16.fnt
vitetrис-0.58.0/src-conf.sh
vitetrис-0.58.0/src/
vitetrис-0.58.0/src/Makefile
vitetrис-0.58.0/src/cfgfile.c
vitetrис-0.58.0/src/cfgfile.h
```

11. EXTRACT THE TARBALL

The file you downloaded has an odd suffix, in fact it's got two; tar and gz. This tells you that it's been encoded and compressed so that its structure and permissions will be preserved after uploading and downloading. Type the following command to extract the file, which is commonly referred to as "the tarball"

`tar xvf v0` (*then just hit tab to auto-complete the rest*)

```
MTM> sudo yum install make gcc
Last metadata expiration check: 0:02:25 ago on Mon 17 Aug 2020 11:23:02 AM EDT.
Dependencies resolved.
=====
Package           Arch   Version        Repository  Size
=====
Installing:
make              s390x  1:4.2.1-10.el8   rhel8-base   496 k
gcc               s390x  8.3.1-5.el8    rhel8-appstream 18 M
Installing dependencies:
pkgconf-pkg-config s390x  1.4.2-1.el8   rhel8-base   15 k
libcrypt-devel     s390x  4.1.1-4.el8   rhel8-base   25 k
pkgconf            s390x  1.4.2-1.el8   rhel8-base   38 k
pkgconf-m4         noarch 1.4.2-1.el8   rhel8-base   17 k
libpkcconf         s390x  1.4.2-1.el8   rhel8-base   34 k
glibc-headers     s390x  2.28-72.el8  rhel8-base   461 k
glibc-devel        s390x  2.28-72.el8  rhel8-base   1.0 M
libatomic          s390x  8.3.1-5.el8   rhel8-base   21 k
libubsan           s390x  8.3.1-5.el8   rhel8-base   143 k
binutils           s390x  2.30-73.el8  rhel8-base   5.9 M
libgomp             s390x  8.3.1-5.el8   rhel8-base   198 k
libasan             s390x  8.3.1-5.el8   rhel8-base   370 k
kernel-headers    s390x  4.18.0-193.14.3.el8_2 rhel8-base   4.0 M
isl                s390x  0.16.1-6.el8   rhel8-appstream 801 k
```

12. INSTALL THE TOOLS

Before we can go much further, we need to install some development tools. Use yum to install **make** and **gcc**

Make is a utility for building software, and gcc is the GNU C Compiler, used to compile software from source. You'll need both.

Some Linux distributions include these types of development tools by default. Other times, it makes sense to leave them out to save disk space. In this case, we have to install them, but it should only take a few seconds.

INSTALLING STUFF IS FUN. WHAT ELSE SHOULD I INSTALL?

So glad you asked.

Most popular software is popular because of how well it works with other software. Once a particular set of software gets well established, it is known as a stack. You may want to look into a LAMP stack. You've already got the first part, Linux and Apache. To complete the stack, look into installing MySQL and PHP.

One of the latest stacks that's gaining in popularity is the MEAN stack, which consists of MongoDB, Express.js, Angular.js, and Node.js. Start reading about these stacks and their variations here: <https://bitbucket.org/blog/lamp-vs-mean-which-stack-is-right-for-you>



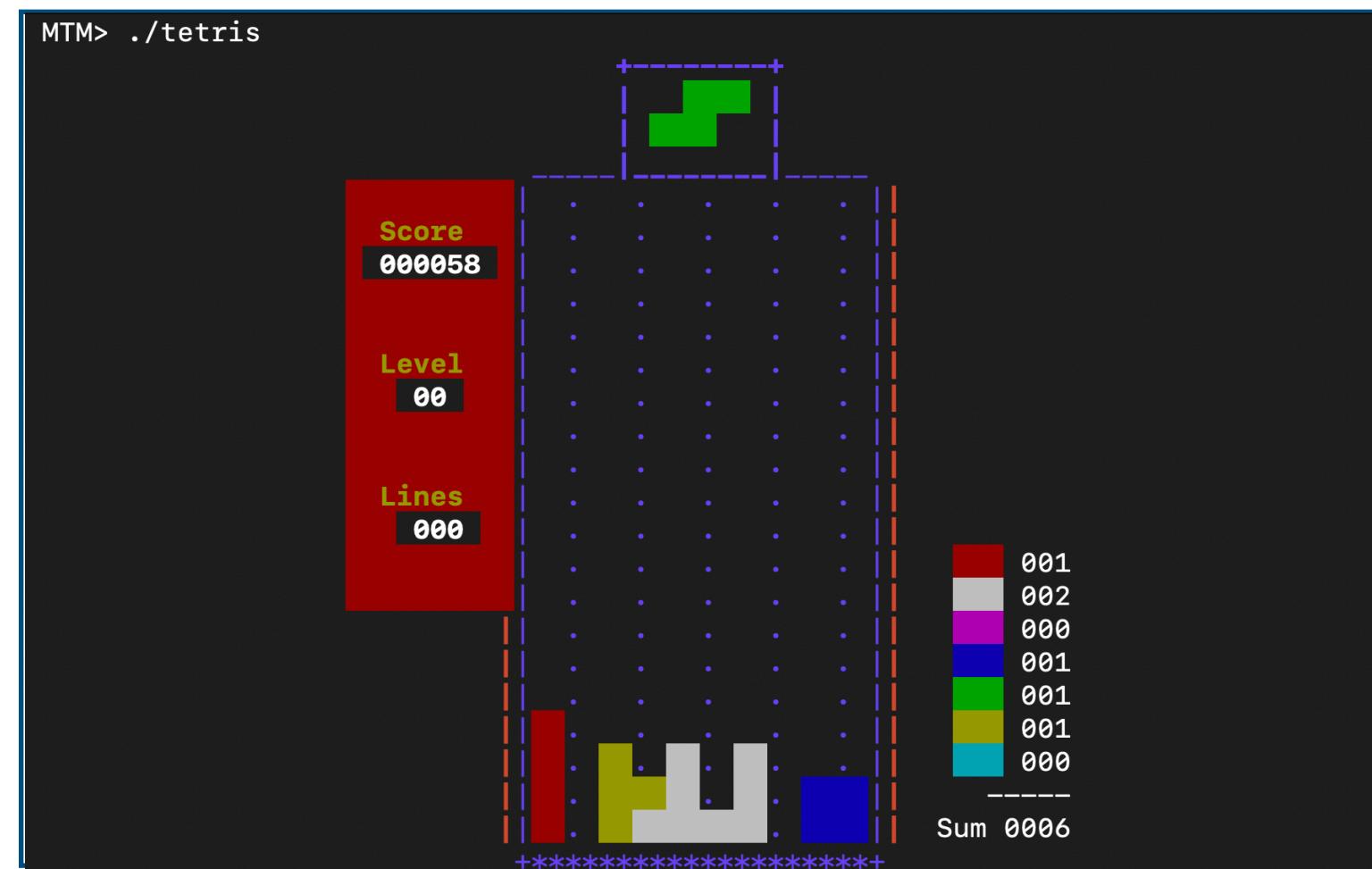
```
MTM> pwd
/home/linux1/vitetrис-0.58.0
MTM> ./configure && make
checking for linux/joystick.h... yes
checking for linux/input.h... yes
checking for Xlib... no
checking for conio.h... no
checking for stdint.h... yes
checking for sys/types.h... yes
checking for sys/select.h... yes
checking for sys/un.h... yes
updating src/config.h
updating config.mk

INSTALLATION DIRECTORIES:
prefix = /usr/local
bindir = $(prefix)/bin
datarootdir = $(prefix)/share
docdir = $(datarootdir)/doc/vitetrис
pixmapdir = $(datarootdir)/pixmaps
desktopdir = $(datarootdir)/applications
```

13. CONFIGURE AND BUILD

Change Directory into the vitetrис directory and type `./configure` to set everything up, and then make to actually build the program from source. Make sure you include the dot-slash before configure or else it may not work.

You can also chain the command together with a pair of ampersands, like I did in the screenshot. This says "Do this, and if it runs without errors, then do this". It will spin for a little bit, and then let you know it's done.



14. VERIFY INSTALLATION

Seems silly to go through all of the trouble of downloading and configuring a game and NOT play it, right? Type `./tetris` and it's game: on.

We could include a lot more text here to help fill up that empty space at the bottom, but you're probably already playing Tetris on a mainframe, so we'll just end there.

```
python3-dnf-plugin-spacewalk.noarch : DNF plugin for Spacewalk
python3-dnf-plugin-spacewalk.noarch : DNF plugin for Spacewalk
python3-dnf-plugin-versionlock.noarch : Version Lock Plugin for DNF
python3-spacewalk-backend-libs.noarch : Spacewalk client tools libraries for Fedora 23
python3-sphinx-theme-alabaster.noarch : Configurable sidebar-enabled Sphinx theme
python3-rhn-virtualization-host.noarch : RHN/Spacewalk Virtualization support specific to Host system
python3-dbus-signature-pyparsing.noarch : Parser for a D-Bus Signature
python3-sphinxcontrib-websupport.noarch : Sphinx API for Web Apps
python3-rhn-virtualization-common.noarch : Files needed by rhn-virtualization-host
python3-keycloak-httpd-client-install.noarch : Tools to configure Apache HTTPD as Keycloak client
python3-dnf-plugin-post-transaction-actions.noarch : Post transaction actions Plugin for python
=====
pygtk2.s390x : Python bindings for GTK+
pygobject2.s390x : Python 2 bindings for GObject
blivet-data.noarch : Data for the blivet python module.
sip.s390x : SIP - Python/C++ Bindings Generator
rhnlib.noarch : Python libraries for the Spacewalk project
rhnlib.noarch : Python libraries for the Spacewalk project
babel.noarch : Tools for internationalizing Python applications
pykickstart.noarch : Python utilities for manipulating kickstart files.
boost-numpy3.s390x : Run-time component of boost numpy library for Python 3
libstoragemgmt-nfs-plugin-clibs.s390x : Python C extension module for libstoragemgmt NFS plugin
libdnf.s390x : Library providing simplified C and Python API to libolv
```

15. YOUR NEXT STEPS

Where you go from here is up to you. You can keep exploring Linux and use this as your go-to system, or maybe keep installing software and maybe build out that web page. If you're going to try and win the whole Master the Mainframe contest (go you!) having a Linux system around can be mighty handy.

You're not required to complete these Linux challenges to complete the contest, so no final validation step is required.

NICE JOB! LET'S RECAP

You may have been asking yourself "I thought this was a mainframe contest... why did I log into another system?"

As you may know, an IBM Z mainframe is able to run many different operating systems at the same time. One of those operating systems is z/OS (which you've been using for the rest of the contest) and it also happens to excel at running Linux. In most big companies, there is a mix of z/OS, Linux, and other operating systems being used. This is Linux on IBM Z hardware, so you're still mastering the mainframe.

NEXT UP...

If you haven't finished one of the other Part 2 challenges, now is the time to go and do that. Otherwise, I'd say you're ready to jump on to Part 3



Want to talk? Join our Slack
ibm.biz/mtm_slack

Tweet about it!
[#MasterTheMainframe](#)