

Lab: Common AOSP Commands

In this lab, we will learn some of the AOSP commands which are pretty useful when you work with the terminal window.

Note:

Initialize the environment with the `envsetup.sh` script:

```
cd ~/aosp  
  
source build/envsetup.sh
```

Here are some of common AOSP commands:

AOSP COMMANDS

gettop	Get the top aosp directory Eg cd \$(gettop)/frameworks/base
croot	Change the current directory to AOSP root
mm	Build the package
cgrep	Only match source files in C/C++ language
jgrep	Same as cgrep, but only matches java source files
resgrep	only matches xml files in the res directory
godir	Goto a directory having the provided file

The first one here is the `gettop` command so this basically prints the top **aosp directory**. Let's quickly try this in the terminal window. Go to the `device/common` folder, type `gettop` that prints the top aosp folder.

```
cd ~/aosp/device/common  
  
gettop
```

```
fenago@fenago:~/aosp$ cd ~/aosp/  
fenago@fenago:~/aosp$ source ./build/envsetup.sh  
fenago@fenago:~/aosp$  
fenago@fenago:~/aosp$  
fenago@fenago:~/aosp$  
fenago@fenago:~/aosp$ cd ~/aosp/device/common  
fenago@fenago:~/aosp/device/common$ gettop  
/home/fenago/aosp  
fenago@fenago:~/aosp/device/common$ █ I
```

Right? From this particular path I am in `device/common` folder, right? If I need to go directly to another folder, go to `frameworks/base`. So I could simply use `cd $(gettop)` will print you the top folder path and from there you put a `/framework/base`.

```
cd $(gettop)/frameworks/base
```

So execute this command, check the path. You are already in the framework/base. So this is a typical use case where you could make use of this `gettop` command. Simply typing `gettop` will bring you top aosp folder. Similarly, if you wanted to go back to `device/common` folder then you can do as follows:

```
cd $(gettop)/device/common
```

Now, let's look at the `croot` command. We can change current directory to aosp root with the help of this command. Type `croot` in the terminal and you should be back in aosp root folder:

```
croot  
  
pwd
```

Let's do another example. Go to `build/blueprint` folder:

```
cd ~/aosp/build/blueprint
```

Type `croot` in the terminal again and you should be back in aosp root folder.

```
croot  
  
pwd
```

Next command is `mm`. So, `mm` command is basically build the package in the current folder. Now, from aosp, let's go to `packages/apps/HTMLViewer` folder:

```
cd ~/aosp/packages/apps/HTMLViewer  
  
ls
```

Let's open `AndroidManifest.xml` and add a new blank line and save it. Now type `mm` command in the terminal, it will only build the HTMLViewer app. From the logs, you can see that it is only building HTMLViewer app.

```

TARGET_ARCH=arm
TARGET_ARCH_VARIANT=armv7-a-neon
TARGET_CPU_VARIANT=generic
HOST_ARCH=x86_64
HOST_2ND_ARCH=x86
HOST_OS=linux
HOST_OS_EXTRA=Linux-5.4.0-105-generic-x86_64-Ubuntu-18.04.3-LTS
HOST_CROSS_OS=windows
HOST_CROSS_ARCH=x86
HOST_CROSS_2ND_ARCH=x86_64
HOST_BUILD_TYPE=release
BUILD_ID=AOSP.MASTER
OUT_DIR=out
PRODUCT_SOONG_NAMESPACES=device/generic/goldfish device/generic/goldfish-opengl
hardware/google/camera hardware/google/camera/devices/EmulatedCamera device/gene
ric/goldfish device/generic/goldfish-opengl
=====
[ 95% 38/40] regenerate globs shard 903 of 1024
0:38 regenerate globs shard 305 of 1024

```

Next, we have `cgrep`, `jgrep` and `resgrep` commands. `cgrep` is only for C/C++ files. `jgrep` is only for java source files and `resgrep` is only for xml files in the res directory.

Let's first use `cgrep` command by typing following command in the terminal:

```
cgrep network
```

Similarly, let's try the `jgrep` command by typing following command in the terminal:

```
jgrep network
```

```

./cts/tests/tests/wifi/src/android/net/wifi/nl80211/cts/PnoNetworkTest.java:95:
o(network);
./cts/tests/tests/wifi/src/android/net/wifi/nl80211/cts/PnoSettingsTest.java:46:
work();
./cts/tests/tests/wifi/src/android/net/wifi/nl80211/cts/PnoSettingsTest.java:47:
s', 's', 'i', 'd' });
./cts/tests/tests/wifi/src/android/net/wifi/nl80211/cts/PnoSettingsTest.java:48:
network1.setFrequenciesMhz(new
int[] { 2412, 2417, 5035 });
./cts/tests/tests/wifi/src/android/net/wifi/nl80211/cts/PnoSettingsTest.java:49:
network1.setHidden(true);
./cts/tests/tests/wifi/src/android/net/wifi/nl80211/cts/PnoSettingsTest.java:51:
PnoNetwork network2 = new PnoNe
work();
./cts/tests/tests/wifi/src/android/net/wifi/nl80211/cts/PnoSettingsTest.java:52:
network2.setSsid(new byte[] { '
a', 's', 'd', 'f' });
./cts/tests/tests/wifi/src/android/net/wifi/nl80211/cts/PnoSettingsTest.java:53:
network2.setFrequenciesMhz(new
int[] { 2422, 2427, 5040 });
./cts/tests/tests/wifi/src/android/net/wifi/nl80211/cts/PnoSettingsTest.java:54:
network2.setHidden(false);
./cts/tests/tests/wifi/src/android/net/wifi/nl80211/cts/PnoSettingsTest.java:56:
return Arrays.asList(network1,
network2);
./cts/tests/tests/wifi/src/android/net/wifi/rtt/cts/TestBase.java:56:    // wait for network selection and connection f
inish
./cts/tools/cts-tradefed/tests/src/com/android/compatibility/common/tradefed/presubmit/CtsConfigLoadingTest.java:91:
    "networking",
./cts/tools/cts-tradefed/tests/src/com/android/compatibility/common/tradefed/presubmit/CtsConfigLoadingTest.java:92:
    "neuralnetworks",
./cts/tools/device-setup/TestDeviceSetup/src/android/tests/getinfo/DeviceInfoConstants.java:40:    public static final
String NETWORK = "network";
./cts/tools/device-setup/TestDeviceSetup/src/android/tests/getinfo/DeviceInfoInstrument.java:123:    // network
./cts/tools/device-setup/TestDeviceSetup/src/android/tests/getinfo/DeviceInfoInstrument.java:124:    String network
= tm.getNetworkOperatorName();
assertThat(network2).isNotEqualT
PnoNetwork network1 = new PnoNe
network1.setSsid(new byte[] { '
network1.setFrequenciesMhz(new
network1.setHidden(true);
PnoNetwork network2 = new PnoNe
network2.setSsid(new byte[] { '
network2.setFrequenciesMhz(new
network2.setHidden(false);
return Arrays.asList(network1,

```

As you can see, these are all java files.

Another useful command is `godir`. It will help you go to a directory having the provided file.

```
godir SystemServer.java
```

```
fenago@fenago:~/aosp$ godir SystemServer.java
fenago@fenago:~/aosp/frameworks/base/services/java/com/android/server$
```

Now, you should be in the directory where `SystemServer.java` exists: `pwd`

From here, if you want to go to `SystemUI.java`, enter following command in the terminal:

```
godir SystemUI.java
```