UNIVERSITY OF THE WEST of SCOTLAND

COMP09024 Unix System Administration

Laboratory 7: System Configuration and Monitoring

System Configuration

Learning Outcomes

- **Understanding the boot process and runlevels.**
- **Understanding how to start and stop system daemons.**
- **Logging in to a remote system using Secure Shell (SSH).**

The first part of this lab is intended to provide an insight into the Linux boot process and the system of runlevels. In the second part we will use SSH, to log in as a different user.

As root begin by taking a snapshot of your current virtual memory status using the **`vmstat`** command to sample the status 5 times at 5 second intervals:

```
[root@UWS /]# vmstat 5 5
```

Wait ~30 seconds until the prompt appears again and make a note of the amount of free memory (shown in the 4th column). Now check the current run level by using the following command:

```
[root@UWS /]# systemctl get-default
```

Check out the available "runlevel" targets:

```
[root@UWS /]# systemctl list-units --type=target
```

Your output should be similar to:

```
 r  b   swpd   free   buff  cache   si   so    bi    bo   in   cs us sy id wa st
 0  0      0 488636      0  12504    0    0     0     0  523   62 39 12 49  0  0
 0  0      0 488512      0  12504    0    0     0     0  113    6  0  0 100  0  0
 0  0      0 488504      0  12492    0    0     0     0  105    4  0  0 100  0  0
 0  0      0 488512      0  12492    0    0     0     0  101    3  0  0 100  0  0
 0  0      0 488504      0  12492    0    0     0     0  105    4  0  0 100  0  0
[root@UWS ~]# systemctl list-units --type=target
UNIT                LOAD   ACTIVE SUB    DESCRIPTION
basic.target        loaded active active Basic System
getty.target        loaded active active Login Prompts
local-fs-pre.target loaded active active Local File Systems (Pre)
local-fs.target     loaded active active Local File Systems
multi-user.target   loaded active active Multi-User System
network.target      loaded active active Network
nss-lookup.target   loaded active active Host and Network Name Lookups
paths.target        loaded active active Paths
remote-fs.target    loaded active active Remote File Systems
slices.target       loaded active active Slices
sockets.target      loaded active active Sockets
swap.target         loaded active active Swap
sysinit.target      loaded active active System Initialization
time-sync.target    loaded active active System Time Synchronized
timers.target       loaded active active Timers

LOAD   = Reflects whether the unit definition was properly loaded.
ACTIVE = The high-level unit activation state, i.e. generalization of SUB.
SUB    = The low-level unit activation state, values depend on unit type.

15 loaded units listed. Pass --all to see loaded but inactive units, too.
To show all installed unit files use 'systemctl list-unit-files'.
[root@UWS ~]#
```

Now change to single user mode or rescue mode:

`[root@UWS ~]# systemctl isolate rescue.target`

Please wait ~20 seconds and continue by giving the *root password for maintenance*: `qwerty`

Question

- **Q7.1)** Describe what happened?

Repeat the **`vmstat`** command again at the prompt:

`[root@UWS ~]# vmstat 5 5`

Your output should be similar to:

```
[root@UWS ~]# systemctl isolate rescue.target
You are in rescue mode. After logging in, type "journalctl -xb" to view
system logs, "systemctl reboot" to reboot, "systemctl default" or "exit"
to boot into default mode.
Give root password for system maintenance
(or type Control-D for normal startup):
sulogin: starting shell for system maintenance
[root@UWS ~]# vmstat 5 5
procs -----------memory---------- ---swap-- -----io---- -system-- ------cpu-----
 r  b   swpd   free   buff  cache   si   so    bi    bo   in   cs us sy id wa st
 0  0      0 491768      0  10896    0    0     0     0  387   45 26  8 66  0  0
 0  0      0 491636      0  10888    0    0     0     0  105    4  0  0 100  0  0
 0  0      0 491644      0  10884    0    0     0     0  101    2  0  0 100  0  0
 0  0      0 491644      0  10884    0    0     0     0  105    3  0  0 100  0  0
 0  0      0 491644      0  10884    0    0     0     0  101    2  0  0 100  0  0
[root@UWS ~]#
```

Question

- **Q7.2)** Is there any difference in the amount of free memory now available, and if yes, how much memory is freed in this mode?

Remaining in this mode, issue a few commands, e.g. **`cd, ls`** and **`pwd`**.

Question

- **Q7.3)** Is there any difference in the output and behaviour of such basic commands?

In this exercise you switched the system in to runlevel 1 (single user mode). This level does not provide mutli-user environment of the mode that you just switched from (runlevel 3 in Linux), but still allows you to perform commands without restriction.

You should have observed that there is small more free memory available, in runlevel 1, because it is less process are running in this runlevel. This mode might be preferable if you are administering a file server, for instance and do not require a graphical interface or if you are investigating a process causing a booting error.

To have a fresh boot, **restart** the Linux by **reloading** the page or by pressing **F5**. Linux will start the boot process and the username and password will be asked again.

Log as a root and type the following command:

```
[root@UWS ~]# init 0
```

and wait for ~ 20 seconds.

Question

- **Q7.4)** What is the outcome of the command **`init 0?`**

The Boot Process

When a PC is booted it usually runs a BIOS program, which is a memory resident program on an EEPROM integrated circuit. The BIOS program will eventually try to read the first sector on a booting media such as a hard-disk. The boot sector on the virtual machine, which emulates a single PC system in our case, contains a small program that the BIOS-equivalent will load and attempt to pass run-control to. This program is called GRUB (Grand Unified Bootloader); one of the most commonly used boot loader programs in Linux (another one is Linux Loader (LILO)). The GRUB configuration is located at **`/boot/grub/grub.cfg`**. This file contains instructions for uploading the kernel and setting the specific options and runlevels. In the labs we will always use the defaults and view the process in the start-up terminal as the system is booting.

The information within **`/boot/grub/grub.cfg`** is read at system start-up by a quickly uploaded mini-linux kernel. As you may have noted the file contains a number of comment-lines that start with a hash ('#') sign.

Now **restart** again the virtualized Linux by **reloading** the page or by pressing **F5**. Linux will start the boot process and the username and password will be asked again.

Questions

- **Q7.5)** How many different *menu entries* can you make out by studying the **`/boot/grub/grub.cfg`** file?

- **Q7.6)** What is the image executable **vmlinuz** for the Linux kernel called? Please provide the absolute pathname.

In the next step, please find the size of the Linux kernel by issuing the command:

```
[root@UWS ~]# ls –lh  /boot/vmlinuz*
```

NB. The * represents a shell wildcard character.

Question

- **Q7.7)** What is the size of the compressed kernel image?

Don't be concerned that the Linux kernel is stored compressed. Actually the kernel contains a small program that is initially invoked to de-compress the kernel itself.

Once de-compressed the bootloader pass the control to the kernel which checks among other things the existence of a video card. Then it checks the hardware (hard-disks, floppies, network adapters, etc), and configures some of its device drivers, while outputting messages about its findings on the start-up terminal (white text on black background).
The output text varies on different systems, depending on the system hardware, the version of Linux being used, and the configuration. For some Unix installations it may be fully suppressed.

In the next step, the kernel will try to mount the root filesystem. The location of the filesystem is configurable in GRUB. The filesystem type is detected automatically. If mounting the root filesystem fails, the kernel will *panic* and halt the system. The root filesystem is usually mounted read-only (abbreviation: '`ro`') so that the filesystem can be checked while it is mounted. This default can be changed using special purpose programs.

Systemd boots towards the target given by "default.target". This is typically a symbolic link to the actual target file. Systemd keeps it's targets in /lib/systemd/system and /etc/systemd/system. A file in /etc/systemd/system takes precedence over those shipped with the OS in /lib/systemd/system -- the intent is that /etc/systemd is used by systems administrators and /lib/systemd is used by distributions.

To list all “runlevel files” (a mapping between runlevels and systemd targets):

```
# ls -l /lib/systemd/system/runlevel?.target
lrwxrwxrwx 1 root root 15 Jul  5 21:31 /lib/systemd/system/runlevel0.target -> poweroff.target
lrwxrwxrwx 1 root root 13 Jul  5 21:31 /lib/systemd/system/runlevel1.target -> rescue.target
lrwxrwxrwx 1 root root 17 Jul  5 21:31 /lib/systemd/system/runlevel2.target -> multi-user.target
lrwxrwxrwx 1 root root 17 Jul  5 21:31 /lib/systemd/system/runlevel3.target -> multi-user.target
lrwxrwxrwx 1 root root 17 Jul  5 21:31 /lib/systemd/system/runlevel4.target -> multi-user.target
lrwxrwxrwx 1 root root 16 Jul  5 21:31 /lib/systemd/system/runlevel5.target -> graphical.target
lrwxrwxrwx 1 root root 13 Jul  5 21:31 /lib/systemd/system/runlevel6.target -> reboot.target
```

Another way to change the runlevel at boot time we just need to create the symbolic link “`default.target`” inside the `/etc/systemd/system/` directory.

Question

- **Q7.8)** What level is chosen to be the default runlevel?

Stop the networking daemon:

```
[root@UWS ~]# systemctl stop systemd-networkd
```

Questions

- **Q7.9)** What is the message displayed if you stop the networking daemon?
- **Q7.10)** What is the output of the **`ping localhost`** command once you have stopped the networking daemon?

Now please check the networking status. To do this you will have to issue the command:

```
[root@UWS ~]# systemctl status systemd-networkd
```

Question

- **Q7.11)** What information was displayed after issuing the previous systemctl command?

Question

- **Q7.12)** Based on your knowledge of the boot process: Draw a diagram that shows the boot process from the very beginning (BIOS, LILO) to runlevel 2. Explain in your own words the complete boot process assuming you have to prepare the diagram for a potential client. Also investigate the standard **System V** runlevel settings and give some reasons why Debian may have deviated from this standard. Research whether the standard Ubuntu, Redhat and Suse distributions adhere to the **System V** definitions of runlevels and note this in your answer.

Remote Administration (via ssh)

As a final exercise we will attempt to connect to another system over the network using the Secure Shell Facility (**ssh**). However, because we are ruining under a web based virtualized system, we are going to connect to our machine rather than a remote one.

Log as student user and check if the **dropear** service is installed and running.

```
[student@UWS ~]# pidof dropbear
```

You should see the **PID** of the **dropbear** server, something similar to:

In the above example, the **129** is the **PID** of the **dropbear** server. If there is no output, the dropbear service is NOT running.

You can also use systemctl to get more information about the status of the process:

```
[student@UWS ~]#  systemctl status dropbear
```

To stop the service:

```
[student@UWS ~]#  systemctl stop dropbear
```

To start again the service:

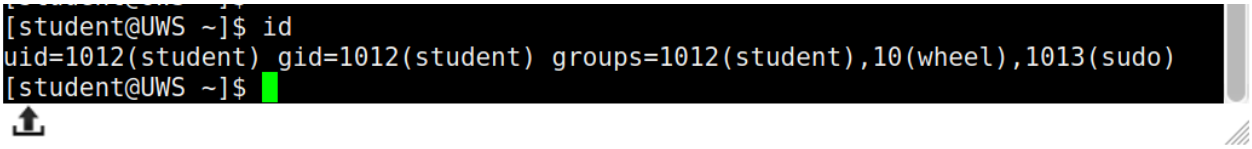
```
[student@UWS ~]#  systemctl stop dropbear
```

Question

- **Q7.13)** Stop and start again the dropbear service and verify that it is running. Check again the PID address and compare it against the previous one. Which PID value is higher?
- **Q7.14)** Why the PID assigned
- **Q7.15)** Is which situation the PID number assigned can be smaller?
- by entering the following to check again for a PID number:

As **student** user, run id command to check you are actually the student user:

[student@UWS ~]# id



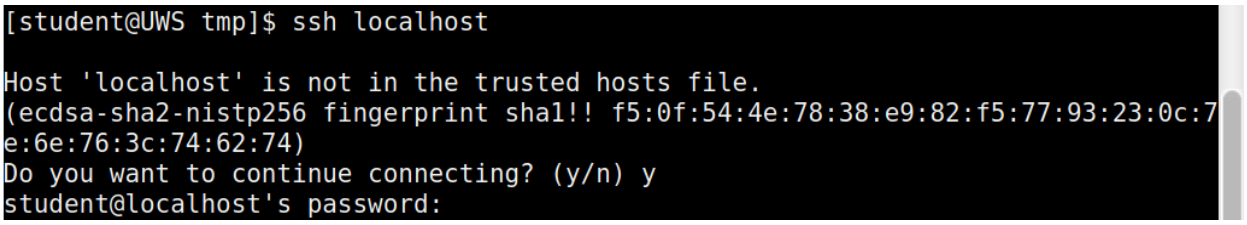
Now, change your directory to *tmp and after that execute the "pwd" command to double check that you are in "*/tmp":

[student@UWS ~]# cd /tmp

[student@UWS tmp]# pwd

Now, stablish a ssh connection against the dropbear server. Press "y" to accept the fingerprint and provide the password.

[student@UWS ~]# ssh localhost



Question

- **Q7.16)** Run the "pwd" command again and indicate the current directory.
- **Q7.17)** Explain why the current directory is not "tmp".

To exit from the SSH connection, type:

[student@UWS ~]# exit

Now, establish a new ssh connection but log in the server as root:

[student@UWS ~]# ssh localhost -l root

Confirm you are by executing the “id” command:

```
[root@UWS ~]# id
```

```
[student@UWS ~]$ ssh localhost -l root
root@localhost's password:
[root@UWS ~]# id
uid=0(root) gid=0(root) groups=0(root),10(wheel)
[root@UWS ~]#
```

– END OF LAB –