# 2 A session with Linux

In this unit, you will go through a longer session with Linux to get to know the system and the terminal a bit better.

From the time you logged on to the system you are “on session” until you log out. As you read this section try those commands yourself in the newly received shell account!

## 2.1 Exploring your system environment

### 2.1.1 Viewing the system date and time

Users can display the system’s current date and time using the date command:

```
$ date
Thu Feb 29 21:21:12 IST 2015
```

The date string above might look strange to you. It is the format originally chosen by the inventors of Linux and still the default format in most of the Linux-like systems. To change the format, give the date command a format string as an option. Here are some examples:

```
$ date “+%T”
21:21:12
$ date “+%Y”
2015
```

Make a note of the time zone in the earlier example. In that IST stands for Indian Standard Time, which is also the time observed in Sri Lanka. IST is 5 h 30 min ahead of UTC, Coordinated Universal Time. Typically, a Linux system’s hardware clock is synchronized to UTC. The system knows the local time from its time zone setting. (And if you carry a computer across time zones, you switch the time zone, not the system clock!)

The original designers of Linux have arbitrarily chosen the instant 00:00:00 UTC of 1 January 1970 as the origin of its system time or the “epoch”. This has become a part of the Linux standard and is widely used. You can get this epoch time in seconds by running `date "+%s"`.

The date command can also change the system time, but only the system administrator is allowed do that. In fact, Linux systems rely much on accurate clock time and therefore almost always synchronized with a time server. But system administration is out of scope right now.

### 2.1.2 Monthly calenders

The standard calender program of Linux is called cal:

```
$ cal
    January 2015
Su Mo Tu We Th Fr Sa
             1  2  3
 4  5  6  7  8  9 10
11 12 13 14 15 16 17
18 19 20 21 22 23 24
25 26 27 28 29 30 31
```

The program `ncal` offers you an alternative layout. Try `cal -3` and ncal -3.

### 2.1.3 How busy is the machine

There is another time-related command called uptime which shows the duration for which the system has been running. In addition, it tells you how many users are currently logged in and prints out three values for system load averages: the load average during the last 1, 5 and 15 minutes.

```
$ uptime
 17:19:20 up 13:34, 6 users, load average: 0.22, 0.18, 0.15
```

The unit of measurement for load average is the number of CPUs being utilized. If the machine is single CPU, 1.00 means the CPU is fully utilized if it has 4 CPUs, a value of 4.00 means that all four CPUs are being fully utilized. (Well, there is more to it. For now, we have to leave it at that.)

There is another famous command called `top` which continuously monitors the system state:

```
top – 17:52:46 up  1:42,  7 users,  load average: 0.10, 0.09, 0.12
Tasks: 157 total,  2 running, 155 sleeping, 0 stopped,   0 zombie
Cpu(s):  1.0 us,  1.1 sy,  0.0 ni, 97.8 id,  0.1 wa,  0.0 hi,  ...
KiB Mem:  7987388 total, 1825888 used, 6161500 free,  79584 buffers
KiB Swap:       0 total,       0 used,       0 free, 854744 cached

  PID USER   PR  NI  VIRT  RES  SHR S  %CPU %MEM    TIME+  COMMAND
 2950 root   20   0  162m  22m 6884 S   5.0  0.3   2:49.21 Xorg
 7829 foo    20   0  416m  34m  16m S   1.0  0.4   0:00.29 termin
 6863 bar    20   0  468m  49m  14m S   0.7  0.6   1:06.15 evince
 [...]
```

### 2.1.4 System information

The system's DNS name can be queried through the command `hostname`. The command uname prints system information like the CPU architecture and the operating system:

```
$ hostname
ce-401
$ uname –a
Linux ce-401 4.13.0-26-generic #29~16.04.2-Ubuntu SMP Tue Jan 9 22:00:44
UTC 2018 x86_64 x86_64 x86_64 GNU/Linux
```

Another source of system information is called environment variables. This is an area in the memory space of the shell where a list of variables and their values are stored. One can query them with the echo command. The value of a variable is addressed by prepending its name by the $-sign:

```
$ echo $OSTYPE
linux–gnu
$ echo $SHELL
/bin/bash
```

Some other useful environment variables are HOSTNAME, HOSTTYPE, TERM, USER. Try them!

Note that traditionally the environment variables are written in all capital.

## 2.2 Screen handling

### 2.2.1 Type-ahead

The terminal emulator reads the keys you type as you type them immediately pass them to the (remote) shell. The keys are normally “echoed” back to the terminal unless they are suppressed as in the case of passwords. If the shell is busy with something else or the connection is very slow, you might not see the echo immediately. But you don’t need to worry. The keys strokes are put to a queue and the shell answers when it is back. Which means you can type blindly ahead of the shell output.

You can think of the keys strokes and the output of the shell as two independent streams of characters. They do not get into the way of the other!

### 2.2.2 Modifying the screen

If too much text is printed on the terminal the `clear` command will clear it. Also, watch the behaviour of the simple Enter key. In fact, it is a simple way of testing whether a shell is responding.

Programs may send special characters to the terminal to get various effects like colour or blinking text. Sometimes this leads to mishaps, where the display gets completely messed up. With the `reset` command, you can recover, but you may have to type it blindly!

The original terminals had fixed sized fonts. Which meant that for a given resolution of the terminal the number of characters per line (called columns) and the number of lines displayed was always the same: 80 columns times 24 lines was a common standard. In today's high-resolution graphics there is no such correlation.

You can query the terminal size through the resize command.

```
$ resize
COLUMNS=92;
LINES=27;
export COLUMNS LINES;
```

### 2.2.3 Control keys

In dedicated terminals, there was a BREAK key to stop whatever the command is running at that moment. Later this function was taken over by the Delete key. In today's keyboards, the Delete key deletes the last character you typed. As you can see these things are system dependent. But a set of commands known as Control commands have a fairly consistent behaviour throughout many systems.

You can break a running program by typing Ctrl+c. To delete a character, the equivalent to Delete, ctrl+h. ctrl+u deletes whole command line to the left of the cursor, ctrl+k kills the part to the right of it. ctrl+d signals the end of input, which will close the shell. Or, if you just want the output to pause, to keep something important disappearing off the screen, type Ctrl+s, the pause. To restart type Ctrl+q.

The following table lists some of the more important Ctrl keys.

| **Shortcut** | **Effect** |
|---|---|
| Ctrl-a | Go to the beginning of the line |
| Ctrl-c | Kill the current process |
| Ctrl-d | (on a line of its own) Exit the current shell (otherwise) delete the character under the cursor |
| Ctrl-e | Go to the end of the line |
| Ctrl-h | Delete the character to the left of the cursor (backspace) |
| Ctrl-k | Delete from the cursor position to the end of line |
| Ctrl-l | Clear the screen |
| Ctrl-u | Delete from beginning of line to cursor position |

### 2.3 More about ending the session

You could have started more shells on top of it. (You just start another shell by typing its name. Try **bash** or **dash**, names of available shells.) Before you log out, you need to close those shells. The proper way to do that is to type ctrl+d (or exit). The "exit" from the last shell will automatically log you out!

**2.4 Assignment 2**

Preparation: When capturing a screen-cast the typical GUI user makes HD films, generating huge files which are sometimes not even sharp. The shell user captures his terminal in the terminal itself. The resulting files are thousand times smaller and always sharp!

This is quite easy: There are many terminal recording tools. We will use `ttyrec`. Just by issuing the command `ttyrec` it will begin capturing everything that happens in the terminal to a file. To end the capture, type exit. You will find the recording in a file called `ttyrecord`. To give it a different name, simply provide the file name as a parameter. For example:

```
$ ttyrec  mywork.ttyrec
[work in the terminal]
$ exit
exit
```

Please be careful not to press non-printable keys like arrows during the recording. They tend to upset the recording!

To play back, run:

```
$ ttyplay mywork.ttyrec
```

**Assignment:** Login to your shell account and first go through the steps given below. Once you are confident capture your work to a file named exxyyy_assignment1.ttyrec. If you still make a mistake, just start ttyrec again with the same file name.

Step 1. Display the system clock time.

Step 2. How many seconds have passed from 00:00:00 UTC of 1 January 1970 to the time you are doing the assignment?

Step 3. Display the output of uptime.

Step 4. Get the values of the environment variables HOSTNAME, HOSTTYPE, USER and HOME?

Step 5. Clear the screen.

Step 6. Find the size of your terminal in columns x lines. Hint: resize.

Now exit the ttyrec. Check the recording by running ttyplay. If everything is OK then go back to the Assignment in Moodle, and upload ttyrec file.

**Summary**

- You know that the system clocks on Linux computers are synchronized to UTC and they calculate the local time from their timezones.
- That the time keeper of Linux ticks relative to the "epoch", which was 00:00:00 UTC on 1 January 1970.
- Simple usage of the commands date, cal and ncal.
- Can get information about the operating system and the working environment from the commands hostname, uname and the environment variables OSTYPE, HOSTNAME, HOSTTYPE, SHELL, TERM.
- You know that the terminals measure their resolution in no. of columns x no. of lines - 80x24 being the standard - and that you can query your terminal program with the resize command.

This handout was derived from the handout prepared by Dr. Visvanath Ratnaweera for the Unix one Course in the year 2017.