

LPIC-1 102-500 – Lesson 8

107.3 Localisation and Internationalisation



Localisation and Internationalisation

- **Localization** is the adaptation of a software from one Language (usually English) to another. The numeronym **L10n** is also used to refer to Localisation.
- **Internationalization** is the capacity of a software to be easily be adapted to other languages The numeronym **i18n** is also used.

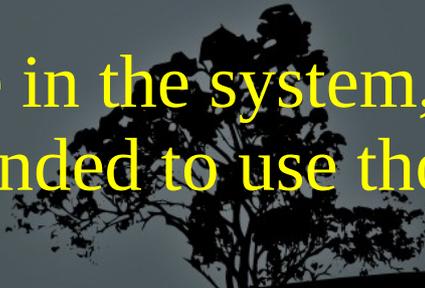


Setting the timezone

- **/etc/localtime**: this is a binary file can be a symlink that points to a timezone file at the **/usr/share/zoneinfo** directory or a copy of one of the timezone files in that directory. This is the way to set the timezone globally in the system.
- **/etc/timezone** # this is a text file that shoes the timezone. Only on Debian.

```
$ cat /etc/timezone  
Asia/Nicosia
```

Note: there are tools for setting the timezone in the system, line the **timedatectl** command and it is recommended to use those than setting the timezone manually.



Show Localisation settings with `locale`

- The **locale** command is used to show information related to localisation.
- **# locale #** show all localisation related shell variables
- **# locale -a #** show all installed localization (locales)



Localisation variables

LANG=en_US.UTF-8 # this variable sets the global localisation of your system

LANGUAGE=el:en_US:en #this variable is a list of localizations on your system that can be read by applications supporting the GNU gettext subsystem.

LC_CTYPE=el_GR.UTF-8 # defines which characters are numeric or alphabetic in each localisation.

LC_NUMERIC="en_US.UTF-8" # sets the delimiters of decials and thousands. For example in the US they use 1,000.00 while in europe 1.000,00.

LC_TIME="en_US.UTF-8" # sets the datetime format

LC_COLLATE=el_GR.UTF-8 # sets the alphabetic priority during sorting

LC_MONETARY="en_US.UTF-8" # sets the currency symbol (\$, €)

LC_MESSAGES=el_GR.UTF-8 # sets the language of system and application settings

LC_PAPER="en_US.UTF-8" # sets the paper size, e.g. letter or A4

LC_NAME="en_US.UTF-8" # sets the names format

LC_ADDRESS="en_US.UTF-8" # sets the address format

LC_TELEPHONE="en_US.UTF-8" # sets the phone numbers format

LC_MEASUREMENT="en_US.UTF-8" # sets the measuring units like metric or imperial.

LC_IDENTIFICATION="en_US.UTF-8" # metadata and information related to localization.

LC_ALL= # This variabl has priority over the other variables. If the is set all the other localisation variables are ignored.

Setting the default localisation

- If we set the localisation variables in `/etc/profile` or, let's say, `/etc/profile.d/lang.sh` this will be the default localisation for all the users in the system. Users can set their own localisation variables using their own `.bash_profile` or `.profile`
- The Default system language can be set with the `localectl` command:

```
# localectl set-locales el_CY.utf8 #  
set the system language to the Greek-  
Cypriot localization.
```



Problems with Localization and LANG=C

- In some situations the user defined localisation may be not installed in the system. For example the “**locale**” to show **en_US.UTF-8** but the “**locale -a**” does not include **en_US.UTF-8** in the installed localisations list. This would create problem when using the shell and when running scripts. Some applications could also misbehave for the same reason.
- The solution is to use a localisation that is always present in the system. Such localisation are **C** and **POSIX**. If we set these values to the **LANG** and **LC_*** variables when running a script or application, we will eliminate any possible issues.
- This happens often when we are connecting to another server (e.g with SSH) and we are using a localisation not installed on the remote system.

The `tzselect` command

- `$ tzselect #` this is a TUI application that can show us the time in another timezone without in fact changing your timezone. In the end it tells you what you need to do to change the timezone permanently



The TZ variable

- You can also set your own timezone to be different from the system using the **TZ** variable:

```
TZ=" :Asia/Nicosia"
```

- To make this persistent you need to set the **TZ** variable in your `~/.profile` or `~/.bash_profile`



Showing the datetime with the `date` command

- The **date** command is used to show or set the date/time on the system but also for displaying the time in various formats.
- **# date #** show the local date and time as defined by `/etc/localtime`
- **# date -u #** show the UTC (Universal Time Coordinated) also known as Global Time.
- **# date +%Z #** show the timezone code of the active timezone.



Character Encodings – ASCII

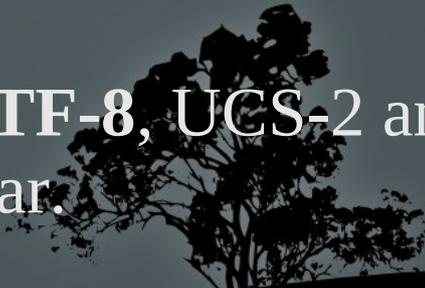
- **ASCII (American Standard Code for Information Interchange)** is a character encoding for the English alphabet and it is composed of 7 bits. This results in 128 characters (2^7), 33 of which are non-printable control characters.
- The problem with ASCII is that it does not support other alphabets like Greek, Cyrillic or even Latin oriented alphabets that contains weird punctuation marks like the French or the German alphabet.
- The '`ascii`' command will print a table with the ASCII characters and their hexadecimal equivalent.



Character Encodings – ISO-8859

- The **ISO-8859** character sequence was an early attempt to internationalise the character encodings. Every character is composed of 8 bits and provides more flexibility than ASCII with a possibility of 256 (2^8) combinations.
- Nevertheless 256 combinations was not be adequate for all languages so there was a different encoding for each writing system. This means that the encoding needs to redefined each time we switch to another system:
 - **ISO-8858-1**: English, German Norwegian, Spanish, etc.
 - **ISO-8859-2**: Serbian, Polish, Czech, Hungarian, etc.
 - **ISO-8859-5**: Cyrillic
 - **ISO-8859-6**: Arabic
 - **ISO-8859-7**: Greek (non polytonic). This is similar (but not identical) to the Windows-1253 encoding.
 - **ISO-8859-9**: Turkish

Character Encodings – Unicode

- **Unicode** is a multibyte standard for the global encoding of characters. The goal was to overcome the 7-8 bit barrier of existing encoding systems like ASCII and ISO-8859. It can use 1 up to 4 bytes per character.
 - The first 127 digits are encoded exactly like ASCII for backward compatibility.
 - It comprises of more than 110.000 characters and more than 100 writing systems, from Greek (including polytonic) and Chinese, even extinct writing systems like Linear B and the ancient Cyprus-Minoan syllabary.
 - Greek characters are represented with two bytes and the Chinese with 3.
 - There are 3 different Unicode encodings. **UTF-8**, **UCS-2** and **UTF-16** . **UTF-8** is by far the most popular.
- 

Character Encodings – UTF-8

- **UTF-8** is an encoding of variable length (1-4 bytes) that can represent every character in the Unicode standard.
- With **UTF-8** you can change your writing system without changing the encoding.
- It is the most popular encoding system with widespread use on all modern operating systems (Linux included) and the World Wide Web.



Convert characters from one encoding to another with `iconv`

- The `iconv` utility can convert text from one encoding to another.
- `# iconv -f WINDOWS-1253 -t ISO-8859-7 greek-1253.txt # convert from Windows-1253 (Windows Greek encoding) to ISO-8859-7 (ISO Greek) and send the output to stdout.`
- `# iconv -f ISO-8859-7 -t UTF-8 greek-8859.txt -o greek-utf8.txt # convert from Greek ISO-8859-7 to Greek UTF-8 and save to the greek-utf8.txt file.`
- `# iconv -c -f UTF-8 -t ISO-8859-7 greek-utf8.txt -o greek-8859.txt # convert from Greek UTF-8 to Greek ISO-8859-7 and save to the greek-8859.txt file, ignoring the non valid characters (-c).`
- `# iconv -l # list of supported encodings.`

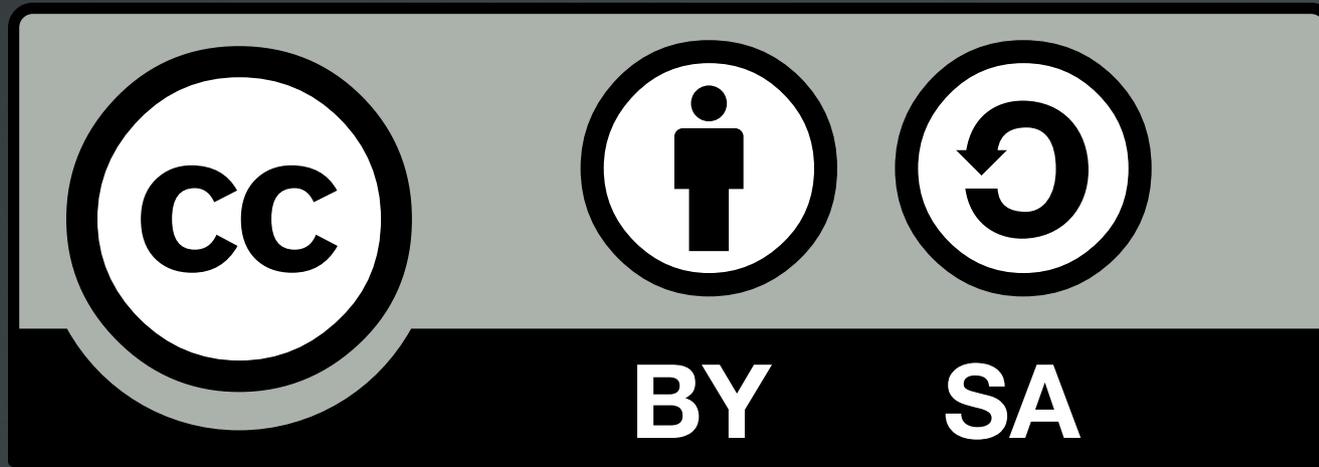


Setting up Greek in your Terminal

- For **RedHat**:
 - `# yum install langpacks-el`
 - `# reboot`
- For **Debian**:
 - `# dpkg-reconfigure locales # select el_GR.UTF-8 and set it as the default encoding`
 - Set the following setting in the `/etc/default/locales` file:
`LANG=el_GR.UTF-8`
`LANGUAGE=el:en_US:en`
 - `# reboot`



License



The work titled "LPIC-1 102-500 – Lesson 8" by Theodotos Andreou is distributed with the Creative Commons Attribution ShareAlike 4.0 International License.

