

LPI 107.3 -Localisation and internationalisation

Curs 2021 - 2022

ASIX M01-ISO 107 Administrative Tasks

Localisation and internationalisation	2
Description	2
Localisation and internationalisation	2
Local environment variables	7
Time / Time Zones / Date	7
Character encoding	10
Example Exercises	12

Localisation and internationalisation

Description

Key concepts:

- ❑ Configure locale settings and environment variables.
- ❑ Configure timezone settings and environment variables.

Commands and files:

- ❑ /etc/timezone
- ❑ /etc/localtime
- ❑ /usr/share/zoneinfo/
- ❑ LC_*
- ❑ LC_ALL
- ❑ LANG
- ❑ TZ
- ❑ /usr/bin/locale
- ❑ tzselect
- ❑ timedatectl
- ❑ date
- ❑ iconv
- ❑ UTF-8
- ❑ ISO-8859
- ❑ ASCII
- ❑ Unicode

Localisation and internationalisation

The concept of localization is to make it easy for the administrator or individual users to set and switch their working environment to match conventions specific to a certain language in a certain country (i.e., Canada/English or Canada/French). A user's locale permits them to interact with system commands, graphical interfaces, and programs naturally without having to translate or convert anything.

The term locale refers to a set of parameters that define the user's language, country, and any special variant preferences. These parameters include the following:

- Language
- Numeric representation
- Date-and-time representation

- Monetary units and symbols
- Case conversion - for proper case mapping of characters
- String collation - for determining sort order rules for a country
- Character classification - determines the correct set of characters, digits, punctuation, and symbols.

Localization is the process of creating or adapting a product to be suitable for a specific group in terms of language, culture, and targeted needs. Locale definition files are used to define the language, territory, and code set information applicable to the user. In a multi-user Linux system, users from different territories may use different languages.

The content of the LANG variable follows the format `ab_CD`, where `ab` is the language code and `CD` is the region code. Locale definition files use the following naming convention:

```
language[_territory][.codeset][@modifiers]
```

```
#1
$ locale
LANG=en_US.UTF-8
LC_CTYPE="en_US.UTF-8"
LC_NUMERIC="en_US.UTF-8"
LC_TIME="en_US.UTF-8"
LC_COLLATE="en_US.UTF-8"
LC_MONETARY="en_US.UTF-8"
LC_MESSAGES="en_US.UTF-8"
LC_PAPER="en_US.UTF-8"
LC_NAME="en_US.UTF-8"
LC_ADDRESS="en_US.UTF-8"
LC_TELEPHONE="en_US.UTF-8"
LC_MEASUREMENT="en_US.UTF-8"
LC_IDENTIFICATION="en_US.UTF-8"
LC_ALL=
```

Commands:

- `locale`
- `locale -c <var>` display the values
- `locale -k <var>` display in key=value format
- `locale -a` display list of all locales

```
#2
$ locale -c LC_NAME
LC_NAME
%d%t%g%t%m%t%f

Mr.
Mrs.
Miss.
Ms.
UTF-8
```

```
$ locale -c LC_TIME
LC_TIME
Sun;Mon;Tue;Wed;Thu;Fri;Sat
Sunday;Monday;Tuesday;Wednesday;Thursday;Friday;Saturday
Jan;Feb;Mar;Apr;May;Jun;Jul;Aug;Sep;Oct;Nov;Dec
January;February;March;April;May;June;July;August;September;October;November;December
AM;PM
%a %d %b %Y %r %Z
```

```
%m/%d/%Y
%r
%I:%M:%S %p
0
S
7
19971130
1
1
2
1
%a %d %b %Y %r %Z
UTF-8
January;February;March;April;May;June;July;August;September;October;November;December
Jan;Feb;Mar;Apr;May;Jun;Jul;Aug;Sep;Oct;Nov;Dec
```

```
$ locale -c LC_NUMERIC
LC_NUMERIC
.
,
3;3
46
44
UTF-8
```

```
#3
$ locale -k LC_NAME
name_fmt="%d%t%g%t%m%t%f"
name_gen=""
name_mr="Mr."
name_mrs="Mrs."
name_miss="Miss."
name_ms="Ms."
name-codeset="UTF-8"
```

```
#4
$ locale -a | head
aa_DJ
aa_DJ.iso88591
aa_DJ.utf8
aa_ER
aa_ER@saaho
aa_ER.utf8
aa_ER.utf8@saaho
aa_ET
aa_ET.utf8
af_ZA
```

```
$ locale -a | grep -i ES
an_ES
an_ES.iso885915
an_ES.utf8
ast_ES
ast_ES.iso885915
ast_ES.utf8
ca_ES
ca_ES@euro
ca_ES.iso88591
ca_ES.iso885915@euro
ca_ES.utf8
ca_ES.utf8@valencia
ca_ES@valencia
eesti
es_AR
es_AR.iso88591
es_AR.utf8
```

Global locale configuration

- /etc/default/locale (Debian-based systems)
- /etc/sysconfig/i18n (Red Hat-based systems)
- /etc/locale.conf

Edit the appropriate file and reboot the system to apply the changes to all the system.

```
#5
$ cat /etc/locale.conf
LANG="en_US.UTF-8"
```

User local configuration

Change the LANG variable value and export it.

```
#6
$ export LANG=ca_Es.UTF-8

$ locale
LANG=ca_Es.UTF-8
LC_CTYPE="ca_Es.UTF-8"
LC_NUMERIC="ca_Es.UTF-8"
LC_TIME="ca_Es.UTF-8"
LC_COLLATE="ca_Es.UTF-8"
LC_MONETARY="ca_Es.UTF-8"
LC_MESSAGES="ca_Es.UTF-8"
LC_PAPER="ca_Es.UTF-8"
LC_NAME="ca_Es.UTF-8"
LC_ADDRESS="ca_Es.UTF-8"
LC_TELEPHONE="ca_Es.UTF-8"
LC_MEASUREMENT="ca_Es.UTF-8"
LC_IDENTIFICATION="ca_Es.UTF-8"
LC_ALL=

$ ls /noexist
ls: no s'ha pogut accedir a '/noexist': El fitxer o directori no existeix
```

LANG=C configuration

Setting the LANG environment variable value to C tells all programs and tools to consider only basic ASCII characters (0-9, A-Z, special characters) and disable UTF-8 multibyte match. It is also used in scripts to predict program output, which may vary based on the current language. In a way, LANG=C disables localization.

```
#7
$ export LANG=C

$ locale
LANG=C
LC_CTYPE="C"
LC_NUMERIC="C"
LC_TIME="C"
LC_COLLATE="C"
```

```
LC_MONETARY="C"
LC_MESSAGES="C"
LC_PAPER="C"
LC_NAME="C"
LC_ADDRESS="C"
LC_TELEPHONE="C"
LC_MEASUREMENT="C"
LC_IDENTIFICATION="C"
LC_ALL=
```

```
$ ls /noexist
ls: cannot access '/noexist': No such file or directory
```

```
$ export LANG=ru_RU.UTF-8
```

```
$ ls /noexist
ls: невозможно получить доступ к '/noexist': Нет такого файла или каталога
```

```
$ LANG=C ls /noexist
ls: cannot access '/noexist': No such file or directory
```

```
#8
$ cat > names
calçots
calamars
california

$ LANG=C sort names
calamars
california
calçots

$ LANG=es_ES.UTF-8 sort names
calamars
calçots
california

$ LANG=ca_ES.UTF-8 sort names
calamars
calçots
california
```

File information

- /usr/lib/locale

```
#9
$ ls /usr/lib/locale/
ca_AD          en_AU.utf8      en_IE@euro      en_US.iso885915 es_CO          es_GT.utf8      es_PY
ca_AD.utf8     en_BW          en_IE.utf8      en_US.utf8      es_CO.utf8     es_HN          es_PY.utf8
ca_ES          en_BW.utf8      en_IL          en_ZA          es_CR          es_HN.utf8     es_SV
ca_ES@euro     en_CA          en_IN          en_ZA.utf8      es_CR.utf8     es_MX          es_SV.utf8
ca_ES.utf8     en_CA.utf8      en_NG          en_ZM          es_CU          es_MX.utf8     es_US
ca_ES@valencia en_DK          en_NZ          en_ZW          es_DO          es_NI          es_US.utf8
ca_FR          en_DK.utf8      en_NZ.utf8     en_ZW.utf8      es_DO.utf8     es_NI.utf8     es_UY
ca_FR.utf8     en_GB          en_PH          es_AR          es_EC          es_PA          es_UY.utf8
ca_IT          en_GB.iso885915 en_PH.utf8     es_AR.utf8      es_EC.utf8     es_PA.utf8     es_VE
ca_IT.utf8     en_GB.utf8      en_SC.utf8     es_BO          es_ES          es_PE          es_VE.utf8
C.utf8         en_HK          en_SG          es_BO.utf8      es_ES@euro     es_PE.utf8     locale-archive
en_AG          en_HK.utf8      en_SG.utf8     es_CL          es_ES.utf8     es_PR          locale-archive.real
en_AU          en_IE          en_US          es_CL.utf8      es_GT          es_PR.utf8
```

```
$ ls /usr/lib/locale/ca_ES@euro/
LC_ADDRESS  LC_CTYPE      LC_MEASUREMENT  LC_MONETARY  LC_NUMERIC  LC_TELEPHONE
LC_COLLATE  LC_IDENTIFICATION  LC_MESSAGES      LC_NAME      LC_PAPER    LC_TIME
```

```
$ file /usr/lib/locale/ca_ES@euro/LC_MONETARY
/usr/lib/locale/ca_ES@euro/LC_MONETARY: glibc locale file LC_MONETARY
```

Local environment variables

The following chart describes environment variables that are typically used to modify locale settings:

LANG

Specifies the default locale to use for attribute categories where neither LC_ALL nor the specific environment variable for that category is set.
Example: LANG="en_US.UTF-8"

LC_ALL

If this environment variable is set, it overrides the selection for all the locales done using the other LC_* environment variables. The value of the other LC_* environment variables is simply ignored in this case.

LC_COLLATE

Specifies what collation order to use for string comparing and sorting.
Example: LC_COLLATE="en_US.UTF-8"

LC_CTYPE

Specifies what locale to use for character sets and character classification.
Example: LC_CTYPE="en_US.UTF-8"

LC_MESSAGES

Specifies what locale to use for printing messages and for parsing responses.
Example: LC_MESSAGES="en_US.UTF-8"

LC_MONETARY

Specifies what locale to use for formatting monetary values. Example:
LC_MONETARY="en_US.UTF-8"

LC_NUMERIC

Specifies what locale to use for formatting numbers. Example:
LC_NUMERIC="en_US.UTF-8"

LC_PAPER

Sets the standard paper size.

LC_TIME

Specifies what locale to use for formatting date/time values. Example:
LC_TIME="en_US.UTF-8"

NLSPATH

Specifies the directories in which the catopen() function looks for message translation catalogs. Example: NLSPATH="/system/nlslib/%N.cat"

It is not mandatory, however, to set the same locale for all variables. It is possible, for example, to have the language defined to pt_BR and the numerical format (LC_NUMERIC) set to the American standard.

Time / Time Zones / Date

Linux (and UNIX) computers keep time in Universal Time (UTC). Since UTC remains constant and is not subject to Daylight Saving Time or other changes, it is useful in synchronizing time across computers and zones.

Linux systems internally keep time using a UTC-synchronized clock that is converted to the appropriate local time based upon user preferences. In a multi-user scenario, users may require different time zones.

The TZ environment variable is used to determine the time zone and how to calculate local time.

- /etc/localtime | /etc/timezone
- /usr/share/zoneinfo
- tzselect (interactive questions to show the timezone name, show not set)
- TZ variable

```
#10
$ ls -l /etc/localtime
lrwxrwxrwx. 1 root root 36 31 oct. 14:20 /etc/localtime ->
../usr/share/zoneinfo/Europe/Andorra

$ ls /usr/share/zoneinfo/
Africa      Brazil     Egypt     GB          Hongkong   Jamaica    MST        Portugal    ROK         UTC
America     Canada    Eire       GB-Eire     HST         Japan      MST7MDT    posix       Singapore   WET
Antarctica  CET       EST        GMT         Iceland    Kwajalein  Navajo     posixrules  Turkey      W-SU
Arctic      Chile     EST5EDT    GMT+0       Indian     leapseconds NZ         PRC         tzdata.zi   zone1970.tab
Asia        CST6CDT   Etc        GMT-0       Iran       Libya      NZ-CHAT    PST8PDT     UCT         zone.tab
Atlantic    Cuba      Europe     GMT0        iso3166.tab MET         Pacific    right       Universal   Zulu
Australia   EET       Factory    Greenwich   Israel     Mexico     Poland     ROC         US
```

tzselect command:

- change the timezone
- TZ variable
- export TZ="Europe/Andorra" (temporary change)
- .bash_profile (TZ='Europe/Andorra'; export TZ)

Change the timezone:

- dpkg-reconfigure tzdata
- redhat-config-date
- system-config-date

Change Time Zone:

- User temporarily setting the TZ variable in the shell session.
- User permanent setting the TZ variable in the user profile ~/.bash_profile.
- System wide: using dpkg-reconfigure tzdata or manually modifying the link.

```
#11
$ tzselect
Please identify a location so that time zone rules can be set correctly.
Please select a continent, ocean, "coord", or "TZ".
 1) Africa
 2) Americas
 3) Antarctica
 4) Asia
 5) Atlantic Ocean
```



```

6) Australia
7) Europe
8) Indian Ocean
9) Pacific Ocean
10) coord - I want to use geographical coordinates.
11) TZ - I want to specify the timezone using the Posix TZ format.
#? 7

```

```

Please select a country whose clocks agree with yours.
1) Albania          14) France          27) Luxembourg      40) Serbia
2) Andorra          15) Germany         28) Malta           41) Slovakia
3) Austria          16) Gibraltar       29) Moldova
42) Slovenia
4) Belarus          17) Greece          30) Monaco           43) Spain
5) Belgium          18) Guernsey         31) Montenegro       44) Svalbard &
Jan Mayen
6) Bosnia & Herzegovina 19) Hungary          32) Netherlands      45) Sweden
7) Britain (UK)      20) Ireland          33) North Macedonia  46)
Switzerland
8) Bulgaria          21) Isle of Man      34) Norway           47) Turkey
9) Croatia            22) Italy             35) Poland            48) Ukraine
10) Czech Republic   23) Jersey            36) Portugal          49) Vatican City
11) Denmark           24) Latvia            37) Romania           50) Åland
Islands
12) Estonia           25) Liechtenstein    38) Russia
13) Finland           26) Lithuania         39) San Marino
#? 2

```

The following information has been given:

Andorra

```

Therefore TZ='Europe/Andorra' will be used.
Selected time is now: Mon Nov  1 16:04:40 CET 2021.
Universal Time is now:      Mon Nov  1 15:04:40 UTC 2021.
Is the above information OK?
1) Yes
2) No
#? 1

```

You can make this change permanent for yourself by appending the line
TZ='Europe/Andorra'; export TZ
to the file '.profile' in your home directory; then log out and log in again.

Here is that TZ value again, this time on standard output so that you
can use the /usr/bin/tzselect command in shell scripts:
Europe/Andorra

```

#12
$ TZ="Europe/Madrid" date
dilluns, 1 de novembre de 2021, 16:12:08 CET

$ TZ="Europe/Moscow" date
dilluns, 1 de novembre de 2021, 18:12:18 MSK

$ TZ="America/Buenos Aires" date
dilluns, 1 de novembre de 2021, 15:12:32 America

$ TZ="America/Buenos_Aires" date
dilluns, 1 de novembre de 2021, 12:14:10 -03

$ TZ="America/Los_Angeles" date
dilluns, 1 de novembre de 2021, 08:14:20 PDT

$ TZ="Australia/Canberra" date
dimarts, 2 de novembre de 2021, 02:14:50 AEDT

$ TZ="Australia/Sidney" date
dilluns, 1 de novembre de 2021, 15:14:59 Australia

```

```
$ TZ="Australia/Sydney" date
dimarts, 2 de novembre de 2021, 02:15:37 AEDT
```

```
#13
# ls -l /etc/localtime
lrwxrwxrwx. 1 root root 36 Oct 31 14:20 /etc/localtime ->
../usr/share/zoneinfo/Europe/Andorra

# rm /etc/localtime
rm: remove symbolic link '/etc/localtime'? y

# ln -s /usr/share/zoneinfo/Europe/Madrid /etc/localtime

# ls -l /etc/localtime
lrwxrwxrwx. 1 root root 33 Nov  1 16:18 /etc/localtime ->
/usr/share/zoneinfo/Europe/Madrid

# timedatectl
    Local time: Mon 2021-11-01 16:19:40 CET
    Universal time: Mon 2021-11-01 15:19:40 UTC
    RTC time: Mon 2021-11-01 15:19:40
    Time zone: Europe/Madrid (CET, +0100)
System clock synchronized: yes
    NTP service: active
    RTC in local TZ: no

# dpkg-reconfigure tzdata
```

Date & timedatectl

These commands are discussed in LPI [108.1-Maintain system time](#).

```
#14
$ date

$ timedatectl

$ date --set 2022-11-01
```

Character encoding

Character encoding is the process of maintaining the mapping between the character and its internal value. In the early stages of computers, a small number of characters were necessary. Fixed width encoding schemes of 4 or 5 digits were sufficient to handle the full set of characters. But as computers advanced in utility and processing, more advanced schemes of encoding were required. Newer schemes include a larger number of digits, and some of them used variable-width encoding.

Characters encoding:

- morse, EBCDIC , Fieldata
- ASCII
- Unicode

- UTF-8
- ISO8859

```
#15
$ locale charmap
UTF-8

$ locale -m
```

ASCII

ASCII (American Standard Code for Information Interchange) is an encoding that is used to represent English language letters, numbers, symbols, and control codes as a 7-bit binary number. The standard ASCII character set includes 128 characters.

```
ASCII(7)                                     Linux Programmer's Manual
ASCII(7)

NAME
    ascii - ASCII character set encoded in octal, decimal, and hexadecimal

DESCRIPTION
    ASCII is the American Standard Code for Information Interchange. It is a
    7-bit code. Many 8-bit
    codes (e.g., ISO 8859-1) contain ASCII as their lower half. The international
    counterpart of ASCII is
    known as ISO 646-IRV.

    The following table contains the 128 ASCII characters.

    C program '\X' escapes are noted.
```

Oct	Dec	Hex	Char	Oct	Dec	Hex	Char
000	0	00	NUL '\0' (null character)	100	64	40	@
001	1	01	SOH (start of heading)	101	65	41	A
002	2	02	STX (start of text)	102	66	42	B
003	3	03	ETX (end of text)	103	67	43	C

Unicode

Unicode is a standard, designed to assign a unique number to every character of every language (including mathematical and other specialized symbols), regardless of the platform and programs being used. Unicode has become the main scheme for internal processing and storage of text in modern computing.

UTF-8

UTF-8 (Unicode Transformation Format – 8-bit) is an encoding that can represent every character in the Unicode character set with 1 to 4 bytes. UTF-8 is backward-compatible with ASCII.

ISO 8859

ISO/IEC 8859 is a joint International Organization for Standardization (ISO) and International Electrotechnical Commission (IEC) series of standards for 8-bit character encodings. Most non-English languages need additional symbols not covered by ASCII. ISO/IEC 8859 solved this problem by utilizing the eighth bit in an 8-bit byte, which allows an additional 96 printable characters to be accommodated.

Conversions between character encodings

- iconv

The iconv command is the standard application programming interface (API) for converting one character encoding to another.

```
iconv -f old-encoding [-t new-encoding] file.txt > newfile.txt
```

```
#16
$ iconv -l
The following list contains all the coded character sets known. This does
not necessarily mean that all combinations of these names can be used for
the FROM and TO command line parameters. One coded character set can be
listed with several different names (aliases).

437, 500, 500V1, 850, 851, 852, 855, 856, 857, 858, 860, 861, 862, 863, 864,
865, 866, 866NAV, 869, 874, 904, 1026, 1046, 1047, 8859_1, 8859_2, 8859_3,
8859_4, 8859_5, 8859_6, 8859_7, 8859_8, 8859_9, 10646-1:1993,
10646-1:1993/UCS4, ANSI_X3.4-1968, ANSI_X3.4-1986, ANSI_X3.4,
ANSI_X3.110-1983, ANSI_X3.110, ARABIC, ARABIC7, ARMSCII-8, ARMSCII8, ASCII,
ASMO-708, ASMO_449, BALTIC, BIG-5, BIG-FIVE, BIG5-HKSCS, BIG5, BIG5HKSCS,

$ iconv -f UTF-8 -t ASCII noms.txt > noms.ascii.txt
```

To reconfigure the keyboard:

```
sudo apt-get install keyboard-configuration
sudo dpkg-reconfigure keyboard-configuration
```

```
sudo apt-get install console-data
sudo dpkg-reconfigure console-data
```

[Debian Reference](#)

To reconfigure Language support:

```
sudo apt-get install locales
sudo dpkg-reconfigure locales

locale-gen
```

Example Exercises

1. Show the locale
2. Change the LANG to english. Do the command `ls /noexist`.
3. Change the LANG to russians. Do the command `ls /noexists`.
4. Change the lang to catalan. Do the command `ls /noexist`.
5. Show the timezone
6. Using `tzselect` establish Europe/Andorra as a time zone.

7. Execute the date command using a on command configuration of the TZ variable in the America/Los_Angeles zone.
8. Repeat the exercise using Australia/Sydney.
9. Again using europe/Moscow.
10. Convert a text file from UTF-8 to ascii.
11. Realitza els exercicis indicats a: [107.3 Localisation and internationalisation](#)
12. Realitza els exercicis del Question-Topics 107.3.