

Los parámetros regionales

1. Introducción

Las distribuciones se instalan y funcionan en numerosos idiomas. Seguramente habrá notado que las páginas de los manuales se muestran en su idioma si se han traducido. Del mismo modo, un gran número de programas muestran su menú en el idioma en el cual se instaló o configuró el sistema.

A diferencia de numerosos editores de programas que ofrecen versiones localizadas diferenciadas de sus programas (por ejemplo, MS Office en español y en inglés son dos versiones distintas), los editores de programas libres suelen integrar directamente un soporte para numerosos idiomas o facilitan paquetes adicionales.

Hay que diferenciar entre la regionalización (en inglés, "localization") y la internacionalización:

- ✦ La **regionalización** consiste en facilitar una traducción de un producto que corresponde a la cultura local (idioma principalmente, pero también moneda o representación de los números). A título de ejemplo, si bien el castellano se utiliza en España, México, Argentina y otros países de América Latina, cada cultura dispone de sintaxis y giros que pueden cambiar, así como de monedas y disposiciones de teclado diferentes.
- ✦ La **internacionalización** prepara la regionalización antes, a nivel de desarrollo del programa. Un solo programa requerirá funciones de una API de internacionalización. Una función encargada de recuperar una cadena de texto en cualquier idioma dentro de una base verificará qué idioma se utiliza y sacará la cadena correspondiente. La librería de funciones `gettext` es un buen ejemplo.

En vez de utilizar un modelo fijado como éste:

```
printf("Hello");
```

un programador procederá así:

```
printf(gettext("Hello"));
```

La función `gettext` buscará la cadena correspondiente a "Hello" en los archivos de regionalización del programa en el idioma actual y sacará una versión traducida. En español, la segunda línea será:

Hola



Para ser precisos, internacionalización se suele escribir `i18n` (i seguido de 18 letras y luego n), y regionalización, `l10n` (l para localization, seguido de 10 letras y luego n).

2. Configuraciones locales

a. Herramientas de la distribución

Hay varias maneras de modificar la regionalización del sistema, incluso de un programa dado. Cada distribución brinda un módulo de configuración, incluso cada entorno de escritorio (Gnome, Unity, KDE...); por ejemplo, un módulo de gestión del idioma, y lo mismo se puede hacer con Debian y Ubuntu de la manera siguiente:

```
dpkg-reconfigure locales
```

Estos métodos sólo modifican parámetros de sistema fácilmente accesibles. Es perfectamente posible modificarlos de forma manual mediante los archivos de configuración de texto.

b. Variables de entorno

Con la consola o en el entorno de escritorio, si se deja por defecto la regionalización, se recupera la información utilizada relativa al país, el idioma o la moneda gracias a unas variables de entorno del shell. La principal se llama **LANG**. Se suele colocar esta variable:

- ~ **/etc/default/locale** en Debian y Ubuntu.
- ~ **/etc/locale.conf** en Red Hat o CentOS.
- ~ **/etc/sysconfig/language** en SuSE.

Su contenido en los equipos del autor es el siguiente:

```
LANG=es_ES.UTF-8
```

Su formato es el siguiente:

```
idioma_Pais[.norma][@variante]
```

- ~ **idioma** es el idioma utilizado (aquí es, para España).
- ~ **País** es el país en mayúscula (aquí ES).
- ~ **.norma** indica la norma utilizada para la codificación de los caracteres (aquí en UTF-8).
- ~ **@variante** precisa variante del idioma. En algunos sistemas que utilizan la codificación ISO-8859-15 se puede especificar @euro para indicar una variante que soporta el euro de esta codificación.

Para pasar al inglés norteamericano:

```
export LANG=en_US.UTF-8
```

El comando **locale** permite recuperar información sobre los elementos de regionalización soportados por su sistema:

```
root@ubuntu:/etc# locale -a | grep -i es
```

```

es_AR.utf8
es_BO.utf8
es_CL.utf8
es_CO.utf8
es_CR.utf8
es_CU
es_CU.utf8
es_DO.utf8
es_EC.utf8
es_ES.utf8
es_GT.utf8
es_HN.utf8
es_MX.utf8
es_NI.utf8
es_PA.utf8
es_PE.utf8
es_PR.utf8
es_PY.utf8
es_SV.utf8
es_US.utf8
es_UY.utf8
es_VE.utf8

```

Los elementos de regionalización soportados por el sistema están en `/usr/share/locale` .

La variable LANG sustituye por defecto a las demás variables de entorno de regionalización, excepto si existen a un nivel inferior. En numerosos Unix, entre los cuales se halla Linux, existe un gran número de variables que empiezan con LC_. Si LANG es la única variable, sustituye a todas estas variables.

Si llama al comando **locale** sin parámetro, muestra lo siguiente:

```

root@ubuntu:/etc# locale
LANG=es_ES.UTF-8
LANGUAGE=es:en_AU:en_CA:en_GB:en_US:en
LC_CTYPE="es_ES.UTF-8"
LC_NUMERIC=es_ES.UTF-8

```

```

LC_TIME=es_ES.UTF-8
LC_COLLATE="es_ES.UTF-8"
LC_MONETARY=es_ES.UTF-8
LC_MESSAGES="es_ES.UTF-8"
LC_PAPER=es_ES.UTF-8
LC_NAME=es_ES.UTF-8
LC_ADDRESS=es_ES.UTF-8
LC_TELEPHONE=es_ES.UTF-8
LC_MEASUREMENT=es_ES.UTF-8
LC_IDENTIFICATION=es_ES.UTF-8
LC_ALL=

```

Se puede modificar y adaptar cada una de las variables LC. Veamos su significado:

- ✧ LC_CTYPE: clase de caracteres y conversión.
- ✧ LC_NUMERIC: formato numérico por defecto, diferente del de la moneda.
- ✧ LC_TIME: formato por defecto de la fecha y la hora.
- ✧ LC_COLLATE: reglas de comparación y de ordenación (por ejemplo, los caracteres acentuados).
- ✧ LC_MONETARY: formato monetario.
- ✧ LC_MESSAGES: formato de los mensajes informativos, interactivos y de diagnóstico.
- ✧ LC_PAPER: formato de papel por defecto (por ejemplo, A4).
- ✧ LC_NAME: formato del nombre de una persona.
- ✧ LC_ADDRESS: igual para una dirección.
- ✧ LC_TELEPHONE: igual para el teléfono.
- ✧ LC_MEASUREMENT: sistema de medidas (métrico en España).
- ✧ LC_IDENTIFICATION: metadatos adicionales referentes a la especificación local.
- ✧ LC_ALL: reglas para todas las demás variables LC.

Veamos un ejemplo con una modificación de LC_TIME:

```
$ date
```

```
mar 11 may 2021 21:06:16 CEST
```

```
$ LC_TIME=en_US date
```

```
Tue May 11 21:06:34 CEST 2021
```



En algunos casos, es necesario exportar la variable `LANG=C` dentro de un script, por ejemplo. Eso tiene como efecto restablecer la regionalización de los comandos, en general en inglés, o de tipo POSIX, con el fin de asegurar un funcionamiento óptimo de los comandos correspondientes.

c. Husos horarios

El huso horario determina el desplazamiento temporal en comparación con la hora universal UTC, permite también gestionar el cambio de hora en primavera y otoño, de forma automática.

Para reconfigurar el huso horario, o **timezone**, emplee los siguientes comandos:

- ▾ Ubuntu y Debian: **dpkg-reconfigure tzdata**, o **timedatectl**
- ▾ Fedora, CentOS y RHEL: **timedatectl** para RHEL7, **system-config-date** hasta RHEL6, o para las anteriores versiones **redhat-config-date**
- ▾ Para los otros: **timedatectl**

Los comandos **tzselect** y **tzconfig** están descontinuados en las distribuciones recientes. En este caso emplee el método manual explicado a continuación, o bien los comandos propios de su distribución.

El comando **timedatectl** funciona en el conjunto de las distribuciones.

El huso horario se determina bajo la forma Continente/ciudad. Esta información se ubica en el archivo **/etc/timezone**.

El archivo **/etc/timezone**, en formato binario, contiene la información vinculada al huso horario, como el desfase horario con respecto a la hora UTC y las reglas para el cambio de

hora en verano o en invierno.

Los archivos de configuración de husos horarios se ubican en **/usr/share/zoneinfo**. Así, `/usr/share/zoneinfo/Europe/Madrid` es copiado o es un enlace simbólico a `/etc/localtime` cuando se modifica el uso horario.

Si existe, la variable **TZ** contiene el huso horario.

```
$ echo $TZ
Europe/Madrid
```

3. Codificación de los caracteres

No todo el mundo utiliza el alfabeto occidental. E incluso en los países occidentales, existen numerosas variantes de un mismo alfabeto. Los ingleses no utilizan acentos, los alemanes tienen caracteres adicionales y otros países emplean más signos auxiliares. Es aún más complicado con los países que no utilizan el mismo alfabeto o que usan ideogramas.

Los nombres de los archivos pueden ser ilegibles en otro país. El texto de un archivo, también.

Cada variante necesita la aplicación de una variante para la codificación de los caracteres. Durante mucho tiempo, la tabla se conocía como ASCII. Se trata de una tabla de caracteres codificada en 1 byte. Por lo tanto, esta tabla puede contener 256 caracteres. Como los primeros 127 son invariables, contiene todos los caracteres americanos, o sea, ningún carácter acentuado, así como los caracteres de control y formateo (supresión, retorno de carro, etc.). Los siguientes 128 son libres. En España se colocan caracteres semigráficos y los acentos, y lo mismo en otros países.

Esto significa que, para cada país, hay que cambiar la tabla ASCII. Además, si accede a una palabra o un nombre de archivo creado en un país y que contiene caracteres propios de este país, hay un riesgo importante de que los nombres ya no se correspondan.

Estas tablas se han normalizado con las normas ISO. Por ejemplo, para España, la tabla ISO 8859-1 corresponde a los caracteres de Europa Occidental en Unix; ISO 8859-15

después del paso al euro. En Windows es la tabla Windows-1252; en DOS, CP850, etc. Cada tabla es incompatible con las demás. Se deben adaptar las fuentes de caracteres para cada tabla. Pero lo peor es que los 256 caracteres no bastan para codificar todos los ideogramas.

Para remediar este problema, existe una norma informática llamada **unicode** compatible con el estándar ISO 10646: da a cualquier carácter de cualquier sistema de escritura de idioma, un nombre y un identificador numérico, y eso de manera unificada, sea cual sea la plataforma informática o el programa (definición inspirada de Wikipedia). Si un sistema acepta unicode y dispone de las fuentes de caracteres asociadas, entonces se visualizan los textos y los nombres de archivos correctamente, sea cual sea el idioma y los caracteres utilizados. Se podrá visualizar correctamente un nombre de archivo en chino en un sistema en francés.

Linux, así como numerosas páginas de Internet, utilizan el formato unicode UTF-8 por defecto. Actualmente, los sistemas más corrientes gestionan correctamente este formato.

Es posible convertir un archivo codificado en una tabla dada hacia otra tabla con el programa **iconv**. El parámetro `-l` le da todas las tablas soportadas. La salida está truncada, ya que ocuparía 4 páginas, hay más de 1200 tablas.

```
# iconv -l
...
437, 500, 500V1, 850, 851, 852, 855, 856, 857, 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,
...
ISO646-ES2, ISO646-FI, ISO646-FR, ISO646-FR1, ISO646-GB, ISO646-HU,
ISO646-IT, ISO646-JP-OCR-B, ISO646-JP, ISO646-KR, ISO646-NO,ISO646-NO2,
...
WINDOWS-31J, WINDOWS-874, WINDOWS-936, WINDOWS-1250, WINDOWS-1251,
WINDOWS-1252, WINDOWS-1253, WINDOWS-1254, WINDOWS-1255, WINDOWS-1256,
WINDOWS-1257, WINDOWS-1258, WINSAMI2, WS2, YU
```

Para convertir un archivo, utilice la sintaxis siguiente:


```
iconv -f WINDOWS-1252 -t UTF8 nombre_archivo
```