PB173 Linux

04 Internacionalizácia

Roman Lacko xlacko1@fi.muni.cz

2022-10-07

Obsah

Internacionalizácia

Internacionalización: Los términos

Internacionalización (i18n)

- Soporte de varios idiomas y hábitos culturales.
- El *locale*.
- Aislamiento de textos traducibles.
- Infraestructura de traducción.

Localización (l10n)

- Textos para el lenguaje concreto.
- Datos para el programa internacionalizado.

Internacionalizácia: Pojmy

Internacionalizácia (i18n)

- Podpora rôznych jazykov a kultúrnych zvyklostí.
- To, čomu systém hovorí „locale“.
- Izolácia preložiteľných dát.
- Infraštruktúra pre preklad.

Lokalizácia (l10n)

- Popis dát pre konkrétny jazyk a kultúrne zvyklosti.
- Dáta pre internacionalizovaný program.

I18N: Prístupy

- Prispôsobenie aplikácie národnému prostrediu.
- Formát čísel, formát dátumu, ...

```
man 7 locale
```

I18N: Premenné prostredia

`LC_COLLATE`	Radenie reťazcov (`strcoll()`)
`LC_CTYPE`	Triedy znakov (`isupper()`, `islower()`)
`LC_MESSAGES`	Jazyk pre správy.
`LC_NUMERIC`	Formát čísel (`printf()`, `scanf()`)
`LC_TIME`	Formát času a dátumu

I18N: Premenné prostredia

Priority

1. `LC_ALL`
2. `LC_*` pre konkrétnu doménu
3. `LANG`

I18N: Prostredie

Špecifikácia nastavenia pre `LC_*` premenné

`LANG[_COUNTRY][.ENCODING][@MODIFIER]`

`LANG` ISO 639-1 (`sk`, `en`, ...)

`COUNTRY` ISO 3316 (`SK`, `GB`, `US`, ...)

`ENCODING` (`UTF-8`, `ISO-8859-2`, ...)

`MODIFIER` Špecifické vlastnosti (`euro`, ...)

I18N: Aplikácia

```
#include <locale.h>
char *setlocale(int category, const char *locale);
```

- `locale.h` definuje makrá `LC_*`
- `locale` môže byť:

  `NULL` Vráti aktuálnu hodnotu.

  `""` Nastaví (a vráti) hodnotu podľa premenných prostredia.

`man 3 uselocale` Nastavenie pre jednotlivé vlákna.

`man 3 nl_langinfo` Zisťovanie vlastností locale.

I18N: Kódovanie

Znaková sada

- Zoznam kódových bodov.
- Mapovanie $\mathbb{N} \rightarrow \Sigma$

Kódovanie

- Konkrétny spôsob uloženia kódových bodov.
- Mapovanie $\mathbb{N} \rightarrow [b_0, b_1, ...]$

I18N: Kódovanie

ASCII

- Názov znakovej sady aj kódovania
- 128 znakov
- Rôzne rozšírenia na 256 znakov
    - ISO-8859-2 (Latin-2, CP852)

Unicode

- Komplexný verzovaný štandard
- 149186 znakov (15.0)
- Niekoľko rôznych kódovaní (UTF-8, UTF-16)

Ako z textového súboru rozoznáme kódovanie?

I18N: UTF-8

Natívne kódovanie v Linux

- K. Thompson, R. Pike
- *Variabilný* počet bajtov na znak
- Kompatibilné s ASCII

```
 7b    0xxx xxxx
11b    110x xxxx  10xx xxxx
16b    1110 xxxx  10xx xxxx  10xx xxxx
21b    1111 0xxx  10xx xxxx  10xx xxxx  10xx xxxx
```

Celkom $2^7 + 2^{11} + 2^{16} + 2^{21} = 2164864$ znakov

I18N: Kódovanie v C

Byte Strings Reťazce bajtov.

Multi-Byte Strings Reťazce znakov vo variabilnom počte bajtov.

Wide Strings Znaky uložené vo viacbajtových poliach.

I18N: Kódovanie v C

`"Řeřicha"`

ISO-8859-2 `d8 65 f8 69 63 68 61 00`

UTF-8 `c5 98 65 c5 99 69 63 68 61 00`

UTF-16 `feff 0158 0065 0159 0069 0063 0068 0061 0000`

I18N: Kódovanie v C

Dĺžka znaku v Multi-Byte kódovaní

```
#include <stdlib.h>
int mblen(const char *s, size_t n);
size_t mbrlen(const char *s, size_t n, mbstate_t *ps);
```

Prevod medzi Multi-Byte a Wide

```
int mbtowc(wchar_t *pwc, const char *s, size_t n);
int wctomb(char *s, wchar_t wc);
```

Vstup a výstup pre Wide reťazce

```
#include <stdio.h>
int wprintf(const wchar_t *format, ...);
int wscanf(const wchar_t *format, ...);
```

I18N: Rozhranie iconv

Prevod reťazcov medzi kódovaniami.
Existuje aj ako program iconv(1).

```
#include <iconv.h>
iconv_t iconv_open(const char *to, const char *from);
void iconv_close(iconv_t cd);

size_t iconv(iconv_t cd, char **inbuf, size_t *inleft,
                         char **outbuf, size_t *outleft);
```

iconv() modifikuje argumenty!

I18N: Dátum a čas

I Am Devloper
@iamdevloper

Elon Musk: I'm putting people on Mars!
Developers: Fantastic, more timezones to support.

2/9/18, 06:03

I18N: Dátum a čas

UNIX epoch

- Počet sekúnd od 1970-01-01 00:00:00 UTC
- Nenesie informáciu o zóne

Koľko hodín je `1570399656`?

```
$ TZ=":Europe/Prague" date "+%F %T" --date @1570399656
2019-10-07 00:07:36

$ TZ=":Asia/Thailand" date "+%F %T" --date @1570399656
2019-10-06 22:07:36
```

I18N: Dátum a čas

```
#include <time.h>
size_t strftime(char *s, size_t max, const char *format,
                const struct tm *tm);
char *strptime(const char *s, const char *format,
                struct tm *tm);
```

Formátovacie značky podobné `sprintf()`

`%c`, `%x`, `%X` Formát pre aktuálnu lokalizáciu.
`nl_langinfo()` s `D_T_FMT`, `D_FMT` resp. `T_FMT`

`%F` Ekvivalent `%Y-%m-%d` (ISO 8601).

`%T` Ekvivalent `%H:%M:%S`.

`%z`, `%Z` Časová zóna.

I18N: Dátum a čas

```c
#include <time.h>
void tzset(void);

extern char *tzname[2];
extern long timezone;
extern int daylight;
```

Nastavenie zóny z premennej prostredia `TZ`

- `std offset[dst[offset]][,start[/time],end[/time]]`
- `:Continent/City`, např. `:Europe/Prague`

Lokalizácia

Lokalizácia

L10N: Katalóg správ

Súbor, ktorý obsahuje preklad aplikácie.

- Rozdelenie práce medzi programátora a prekladateľa.
- Pridanie lokalizácie bez zásahu do kódu.

Portable Object (PO)

Textový súbor s prekladom.
Jednoduchšie na editáciu prekladateľom.

Machine Object (MO)

Binárna podoba PO pre aplikáciu.
Efektívnejší prístup k prekladu.

L10N: gettext()

Nastavenie zdroja prekladu

```
#include <libintl.h>
char *bindtextdomain(const char *domain, const char *dirname);
char *textdomain(const char *domain);
```

Preklad

```
char *gettext(const char *msgid);
char *ngettext(const char *msgid, const char *plural,
        unsigned long n);
#define _(STR) gettext(STR)
```

Zdroje

Zdroje

- GetText manuál
- Unicode 15