

Lokalizacja projektów w Linuxie

Gettext

Autor: Jakub Ejnik

Co to jest Gettext ?

Gettext to pakiet oprogramowania , stworzony w ramach projektu GNU, służący do internacjonalizacji programów komputerowych na uniksopodobnych systemach operacyjnych. Najpopularniejsza implementacja gettext jest rozwijana przez Projekt GNU w ramach wolnego oprogramowania

Kto, gdzie i kiedy ?

Gettext został stworzony w Sun Microsystems na początku lat 90. XX w. Wersja stworzona przez Projekt GNU została wydana w 1995 roku.

Z czym to się je ?

Tak jak już wcześniej to było wspomniane gettext to pakiet oprogramowania służący do internacjonalizacji tudzież lokalizacji programów. Tłumacząc to na bardziej zrozumiały język gettext to tak naprawdę pakiet oprogramowania, który wprowadza do programu możliwości obsługi wyświetlania komunikatów i dokumentacji w wielu różnych językach

Gettext w praktyce

```
#include<libintl.h>
#include<locale.h>
#include<stdio.h>

#define _(String) gettext (String)

int main()
{
    setlocale(LC_ALL, "");
    bindtextdomain("helloworld", "/usr/share/locale");
    textdomain("helloworld");
    printf(_("Hello World\n"));
    return 0;
}
```

Gettext w praktyce

- Następnie tworzymy katalog np. `mkdir po/`
- Po czym wypakowujemy do pliku o rozszerzeniu `.pot`
`xgettext -d helloworld -o po/helloworld.pot -k_ -s helloworld.c`

Gettext w praktyce

```
# SOME DESCRIPTIVE TITLE.
# Copyright (C) YEAR THE PACKAGE'S COPYRIGHT HOLDER
# This file is distributed under the same license as the PACKAGE package.
# FIRST AUTHOR <EMAIL@ADDRESS>, YEAR.
#
#, fuzzy
msgid ""
msgstr ""
"Project-Id-Version: PACKAGE VERSION\n"
"Report-Msgid-Bugs-To: \n"
"POT-Creation-Date: 2010-04-27 17:42+0530\n"
"PO-Revision-Date: YEAR-MO-DA HO:MI+ZONE\n"
"Last-Translator: FULL NAME <EMAIL@ADDRESS>\n"
"Language-Team: LANGUAGE <LL@li.org>\n"
"MIME-Version: 1.0\n"
"Content-Type: text/plain; charset=CHARSET\n"
"Content-Transfer-Encoding: 8bit\n"

#: helloworld.c:13
#, c-format
msgid "Hello World\n"
msgstr ""
```

Gettext w praktyce

- Tworzymy katalog który będzie zawierał nasze tłumaczenie

`mkdir hi/`

- Po czym zmieniamy rozszerzenie z .pot na po

`cp helloworld.pot hi/helloworld.po`

Gettext w praktyce

```
# Hello World Localization.
# Copyright (C) 2010 Naveen Kumar
# This file is distributed under the same license as the PACKAGE package.
# Naveen Kumar <nkumar@redhat.com>, 2010.
#
msgid ""
msgstr ""
"Project-Id-Version: helloworld 1.0\n"
"Report-Msgid-Bugs-To: \n"
"POT-Creation-Date: 2010-04-27 18:31+0530\n"
"PO-Revision-Date: 2010-04-27 18:53+0530\n"
"Last-Translator: Naveen Kumar <nkumar@redhat.com>\n"
"Language-Team: Hindi <LL@li.org>\n"
"MIME-Version: 1.0\n"
"Content-Type: text/plain; charset=UTF-8\n"
"Content-Transfer-Encoding: 8bit\n"

#: helloworld.c:13
#, c-format
msgid "Hello World\n"
msgstr "नमस्कार दुनिया\n"
```

Gettext w praktyce

- Kolejnym krokiem jest konwersja do .mo
`msgfmt helloworld.po -o helloworld.mo`
- Po czym należy przenieść plik .mo do
`/usr/share/locale/hi/LC_MESSAGES` gdzie domyślnie lądują pliki translacyjne
`cp helloworld.mo /usr/share/locale/hi/LC_MESSAGES/`
- Kompilujemy nasz plik `gcc -o helloworld helloworld.c`
- Następnie uruchamiamy skompilowany plik
`LANG=hi_IN ./helloworld`

Gettext

Co w efekcie skutkuje wyświetleniem przetłumaczonego komunikatu

```
[user@localhost]$ ./helloworld
```

नमस्कार दुनिया ← Hello Wrold w hindi ;)

KONIEC

Dziękuję za uwagę