

# ПРОГРАММИРОВАНИЕ В СРЕДЕ UNIX

---

Библиотека PCRE



Кафедра «Управления информационными системами и  
программирования»  
Иванов Е.А. 2017

# Библиотека PCRE

---

- Ранее мы рассмотрели REGEX для работы с регулярными выражениями, однако, REGEX не позволяет работать с различными кодировками, предоставляет достаточно ограниченные возможности по сравнению с теми, что предоставляет язык PERL.
- Разработчик программ на C++ может воспользоваться библиотекой Perl Compatible Regular Expressions (PCRE), ее исходные коды и документация находятся на сайте <http://www.pcre.org>.

# Библиотека PCRE

---

- Библиотека PCRE – это набор функций реализующих работу с регулярными выражениями в формате языка Perl 5.
- Данная библиотека была разработана для почтового сервера Exim, версия 1.0 была выпущена в ноябре 1997.
- На сегодняшний день данная библиотека используется в таких открытых продуктах, как Apache web server, PHP, Postfix, и т. д.
- Библиотека доступна для операционных систем POSIX(Linux/Unix), Mac OSX, Windows.

# Библиотека PCRE

---

- Для начала работы с библиотекой нужно подключить заголовочный файл **pcre.h**
- При компиляции нужно указывать опцию `-I pcre` (или `-Ipcre`)

Пример:

```
gcc pcredemo.c -Ipcre -o pcredemo
```

- В Windows нужно к проекту подключить `pcre.lib` или `pcre.dll`

# Библиотека PCRE

---

## Пример на Си

```
#include <stdio.h>
#include <string.h>
#include "pcre.h"
#define OVECCOUNT 30 /* должно быть кратно трем */
int main(int argc, char* argv[])
{
    pcre *re;
    const char *error;
    int erroffset;
    int ovector[OVECCOUNT];
    int rc;
```

# Библиотека PCRE

---

## Пример на Си (продолжение)

// Задаем свой шаблон

```
char *regex = "^From: ([^@]+)@([^\r]+)";
```

// Определяем строку в которой будем искать

```
char *data = "From: regular.expressions@example.com\r\n";
```

# Библиотека PCRE

---

## Пример на Си (продолжение)

```
// Компилируем выражение
```

```
re = pcre_compile(
    regex,      /* the pattern */
    0,         /* default options */
    &error,     /* for error message */
    &erroffset, /* for error offset */
    NULL);    /* use default character table */
if (! re)
{
    fprintf(stderr, "PCRE compilation failed at expression offset %d: %s\n",
        erroffset, error);
    return 1;
}
```

# Библиотека PCRE

---

## Пример на Си (продолжение)

```
// Выполняем поиск
```

```
rc = pcre_exec(  
    re, /* the compiled pattern */  
    NULL, /* no extra data - we didn't study the pattern */  
    data, /* the subject string */  
    strlen(data), /* the length of the subject */  
    0, /* start at offset 0 in the subject */  
    0, /* default options */  
    ovector, /* output vector for substring information */  
    OVECCOUNT); /* number of elements in the output vector */
```



# Библиотека PCRE

---

## Пример на Си (продолжение)

```
// Обработка ошибок поиска
if (rc < 0) {
    switch(rc) {
        case PCRE_ERROR_NOMATCH:
            printf("No match found in text\n");
            break;
        /* More cases defined... */
        default:
            printf("Match error %d\n", rc);
            break;
    }
    return 1;
}
```

# Библиотека PCRE

---

## Пример на Си (продолжение)

```
// Получение найденных подстрок
if (rc < 3) {
    printf("Match did not catch all the groups\n");
    return 1;
}
/* ovector[0]..ovector[1] are the entire matched string*/
char *name_start = data + ovector[2];
int name_length = ovector[3] - ovector[2];
char *domain_start = data + ovector[4];
int domain_length = ovector[5] - ovector[4];
```

# Библиотека PCRE

---

## Пример на Си (продолжение)

```
// Получение найденных подстрок
/* Finally, print the match */
printf("Mail from: %.*s domain: %.*s\n",
name_length, name_start, domain_length, domain_start);
return 0;
} //END main
```

# Библиотека PCRE

---

PCRE сохраняет индексы найденных подстрок в массив следующего формата

```
+-----+
| Match 0 | Match 0 | Match 1 | Match 1 | ... |
| start   | end     | start   | end     |   |
+-----+
```

Количество найденных подстрок содержится в `rc`

# Библиотека PCRE

---

Существует так же библиотека PCRE для C++  
pcrecpp

# Дополнительные материалы

<http://www.regular-expressions.info/pcre.html>

<http://haifux.org/lectures/156/PCRE-Perl-Compatible-Regular-Expression-Library.pdf>

<http://www.shtogrin.com/library/web/pcre/doc/pcre/>