Операционные системы и сети

Язык программирования C

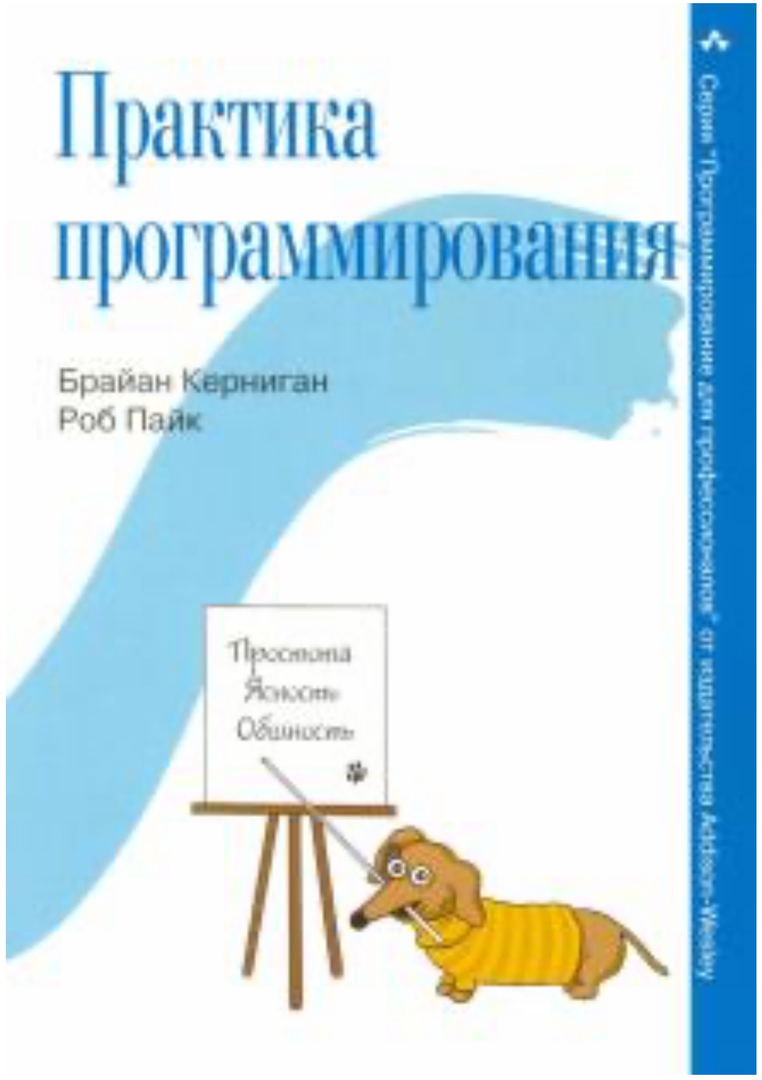
Павел Филонов 28/01/2012

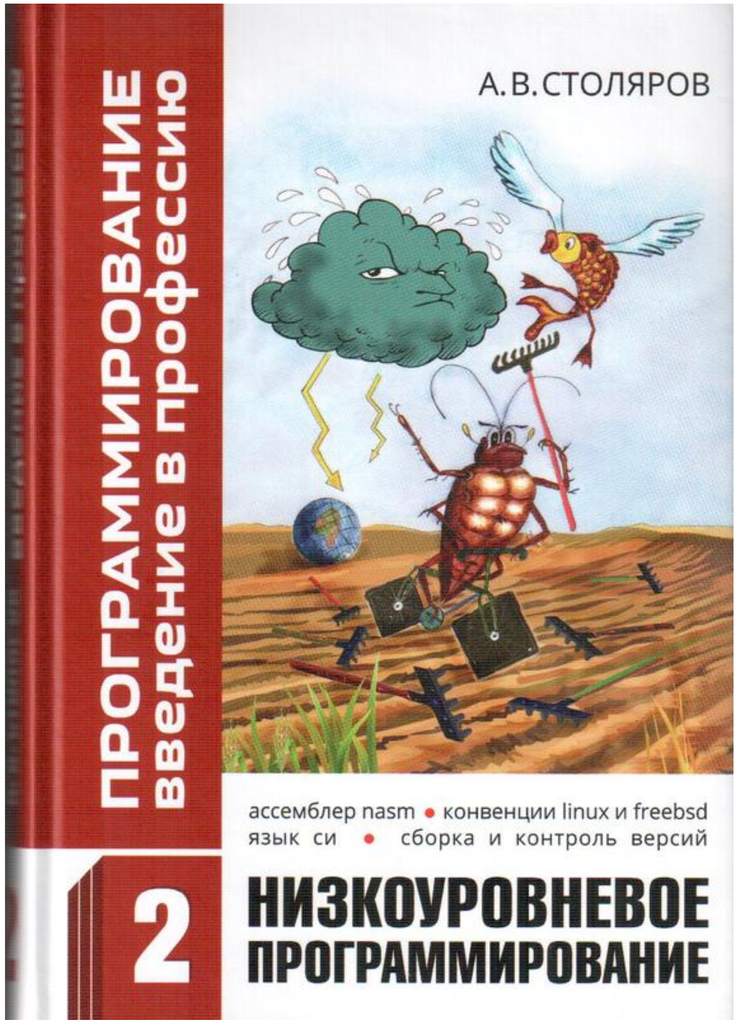
Краткое описание

- Цель создания - портирование ОС UNIX на различные аппаратные платформы
- Год создания - 1972, автор - Денис Ритчи
- Последняя версия стандарта языка - 2018 год (C17)
- Популярные компиляторы
  - GCC
  - Clang
  - Microsoft Visual C++ (MSVC)

Литература


ВТОРОЕ ИЗДАНИЕ
ЯЗЫК ПРОГРАММИРОВАНИЯ
ANSI C
БРАЙАН КЕРНИГАН
ДЕННИС РИТЧИ
Серия книг по программированию от Prentice Hall


Практика программирования
Брайан Керниган
Роб Пайк
Простота
Ясность
Общность


А.В.СТОЛЯРОВ
ПРОГРАММИРОВАНИЕ
введение в профессию
2
ассемблер nasm • конвенции linux и freebsd
язык си • сборка и контроль версий
НИЗКОУРОВНЕВОЕ ПРОГРАММИРОВАНИЕ

Hello, world!

```c
/* Prints "Hello, world!" */

#include <stdio.h> /* declaration of printf func */

int main() /* program entry point */
{
    printf("Hello, world!\n"); /* '\n' - special symbol for newline */

    return 0; /* zero treats as success return code */
}
```

Компиляция и компоновка:
gcc hello.c -o hello

Запуск
./hello

Полезные опции компиляции:
gcc -Wall -g hello.c -o hello

Пример. Фаренгейт в Цельсий

```c
/* print Fahrenheit-Celsius table
    for fahr = 0, 20, ..., 300 */

int main()
{
    int fahr, celsius;
    int lower, upper, step;

    lower = 0;      /* lower limit of temperature table */
    upper = 300;    /* upper limit */
    step = 20;      /* step size */

    fahr = lower;

    while (fahr <= upper) {
        celsius = 5.0/9.0 * (fahr - 32.0);
        printf("%d\t%d\n", fahr, celsius);
        fahr = fahr + step;
    }

    return 0;
}
```

Пример. Подсчет символов, строк и слов

```c
/* count lines, words, and characters in input */

#include <stdio.h>

#define IN   1   /* inside a word */
#define OUT  0   /* outside a word */

int main()
{
    int c, nl, nw, nc, state;

    state = OUT;
    nl = nw = nc = 0;

    while ((c = getchar()) != EOF) {
        ++nc;
        if (c == ´\n´)
            ++nl;
        if (c == ´ ´ || c == ´\n´ || c == ´\t´)
            state = OUT;
        else if (state == OUT) {
            state = IN;
            ++nw;
        }
    }
    printf("%d %d %d\n", nl, nw, nc);
    return 0;
}
```

Пример. Подсчет цифр

```c
/* count digits, white space, others */

#include <stdio.h>

int main()
{
    int c, i, nwhite, nother;
    int ndigit[10];

    nwhite = nother = 0;
    for (i = 0; i < 10; ++i)
        ndigit[i] = 0;

    while ((c = getchar()) != EOF)
        if (c >= '0' && c <= '9')
            ++ndigit[c-'0'];
        else if (c == ' ' || c == '\n' || c == '\t')
            ++nwhite;
        else
            ++nother;

    printf("digits =");
    for (i = 0; i < 10; ++i)
        printf(" %d", ndigit[i]);
    printf(", white space = %d, other = %d\n", nwhite, nother);

    return 0;
}
```

Пример. Возведение в степень

```c
/* test power function */

#include <stdio.h>

int power(int m, int n); /* declaration */

int main()
{
    int i;

    for (i = 0; i < 10; ++i)
        printf("%d %d %d\n", i, power(2,i), power(-3,i));

    return 0;
}

/* power:  raise base to n-th power; n >= 0 */
int power(int base, int n)
{
    int i, p;
    p = 1;

    for (i = 1; i <= n; ++i)
        p = p * base;

    return p;
}
```

Пример. Строки

```c
/* print longest input line */

#include <stdio.h>

#define MAXLINE 1000    /* maximum input line size */

int getline(char line[], int maxline);
void copy(char to[], char from[]);

int main()
{
    int len;            /* current line length */
    int max;            /* maximum length seen so far */
    char line[MAXLINE];     /* current input line */
    char longest[MAXLINE];  /* longest line saved here */

    max = 0;

    while ((len = getline(line, MAXLINE)) > 0)
        if (len > max) {
            max = len;
            copy(longest, line);
        }

    if (max > 0)    /* there was a line */
        printf("%s", longest);

    return 0;
```

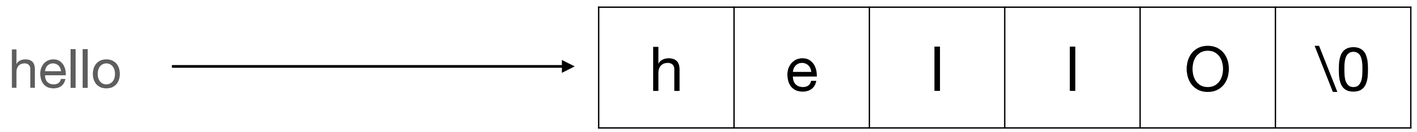
Пример. Строки (продолжение)

```c
/* getline:  read a line into s, return length */
int getline(char s[], int lim)
{
    int c, i;

    for (i = 0; i < lim - 1 && (c = getchar()) != EOF && c != '\n'; ++i)
        s[i] = c;

    if (c == '\n') {
        s[i] = c;
        ++i;
    }

    s[i] = '\0';

    return i;
}


/* copy:  copy 'from' into 'to'; assume to is big enough */
void copy(char to[], char from[])
{
    int i;
    i = 0;

    while ((to[i] = from[i]) != '\0')
        ++i;
}
```

Строки в языке C

- Строки представляют собой массивы символов
- Строки оканчиваются специальным символом ‘\0’ с ASCII кодом 0

```
char hello[] = "hello";
```



Типы данных

- char - 1 байт. Подходит для хранения кодов ASCII символов
- int - целое число со знаком (размер зависит от разрядности процессора. Обычно 4 байта)
- float - число с плавающей точкой одинарной точности (4 байта)
- double - число с плавающей точкой двойной точности (8 байт)

Спецификаторы типа:

- short int - короткое целое (обычно 2 байта)
- long int - длинное целое (обычно 4 или 8 байт)
- unsigned int - беззнаковое целое размера int
- signed int - знаковое целое размера int (по умолчанию все целые числа - знаковые)

Константы

- 1234 - int
- 1234L - long
- 123u - unsigned int
- 123ul - unsigned long int
- 1.12 - double
- 1.12f - float
- 1e-02 - double
- 0123 - восьмиричная СС
- 0xFA - шестнадцатеричная СС
- ‘A’ - char
- “Hello” - char[]

Перечисления

```
enum boolean { NO, YES };
                /* NO is 0, YES is 1 */

enum escapes { BELL = '\a', BACKSPACE = '\b', TAB = '\t',

               NEWLINE = '\n', VTAB = '\v', RETURN = '\r' };

enum months { JAN = 1, FEB, MAR, APR, MAY, JUN,
              JUL, AUG, SEP, OCT, NOV, DEC };
                    /* FEB is 2, MAR is 3, etc. */
```

Операторы

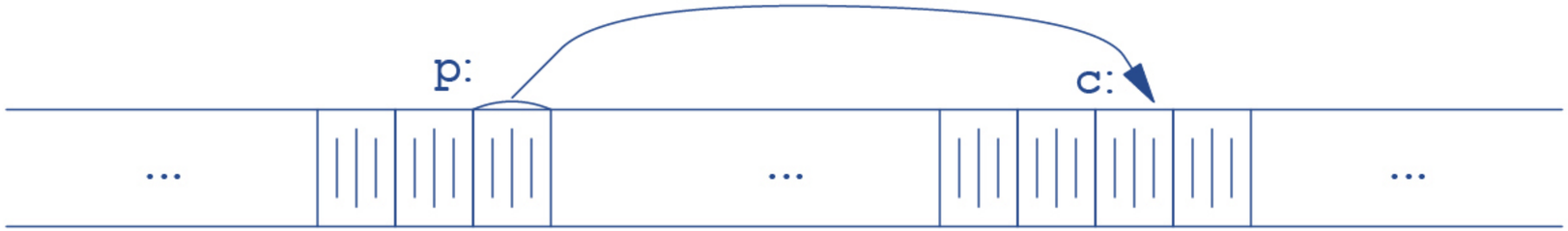
- Арифметические: +, -, *, /, % (остаток), ++ (инкремент), -- (декремент)
- Сравнения: <, >, <=, >=, == (равно), != (не равно)
- Логические: && (AND), || (OR), ! (NOT)
- Побитовые: & (AND), | (OR), ^ (XOR), >> (левый сдвиг), <<(правый сдвиг), ~(инверсия)

Поток управления

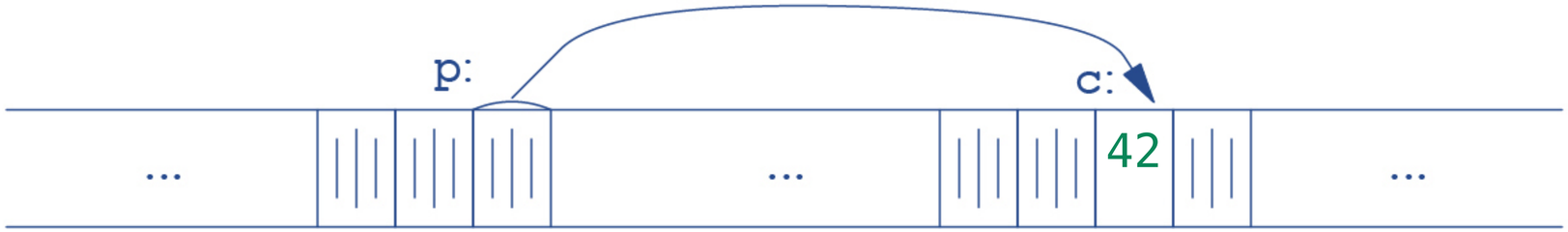
- Ветвление
  - If, if-else, switch
- Циклы
  - while, for, do … while
  - break, continue

Указатели

```
int c;
int *p;
p = &c;
```



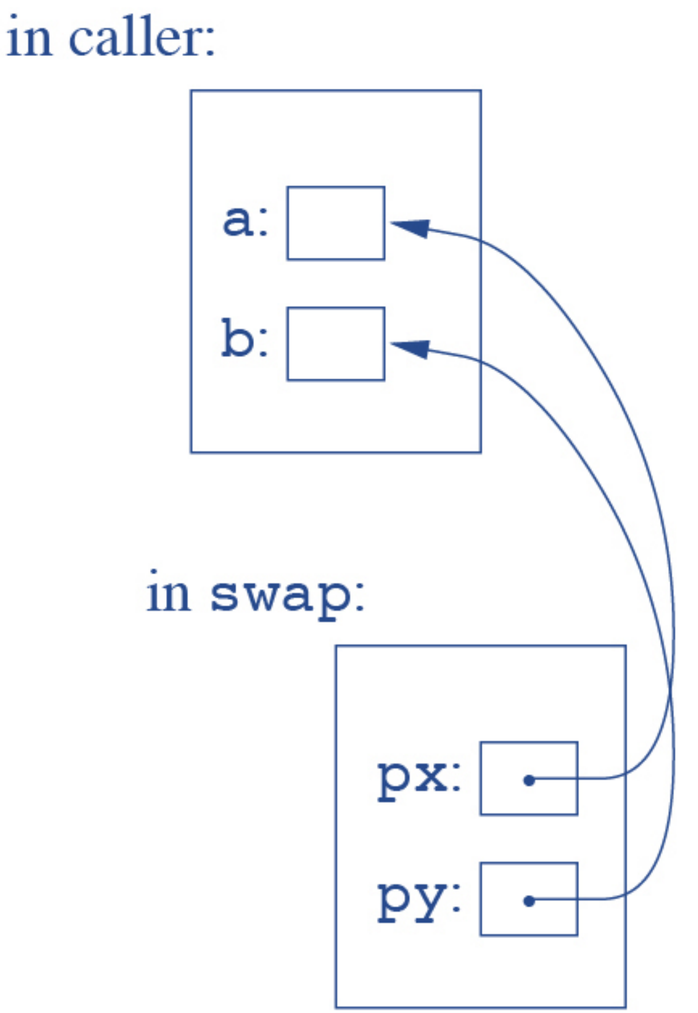

```
*p = 42;
```




```
p = NULL; /* empty pointer. Points to nothing */
```

Пример. Swap

```c
void swap(int *px, int *py)  /* interchange *px and *py */
{
    int temp;

    temp = *px;
    *px = *py;
    *py = temp;
}

int main(){
    int a = 24;
    int b = 42;

    swap(&a, &b);
}
```

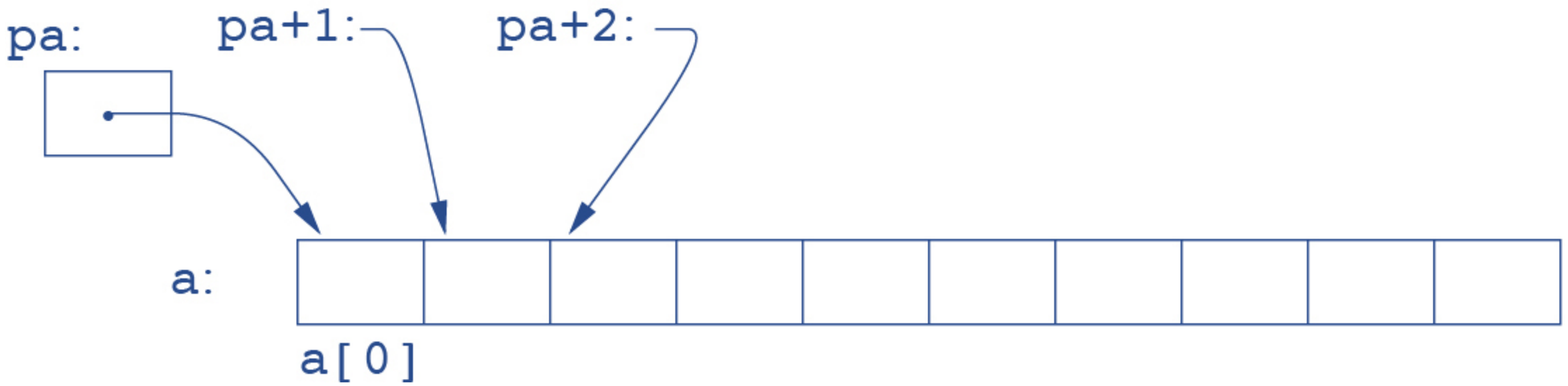


Указатели и массивы

```
int a[10];
```


```
int *pa;
pa = &a[0];
```

Указатели и массивы



Пример. strlen

```
/* strlen:  return length of string s */
int strlen(char *s)
{
    int n;

    for (n = 0; *s != '\0'; s++)
        n++;

    return n;
}
```

Пример. strcpy

```
/* strcpy:  copy t to s; pointer version 1 */
void strcpy(char *s, char *t)
{
    while ((*s = *t) != '\0') {
        s++;
        t++;
    }
}
```

```
/* strcpy:  copy t to s; pointer version 2 */
void strcpy(char *s, char *t)
{
    while (*s++ = *t++)
        ;
}
```

Структуры

```
struct point {
    int x;
    int y;
};

struct point pt;
pt.x = 10;
pt.y = 20;

struct point *ppt = &pt;
ppt->x = 20;
ppt->y = 20;
```

Рекурсивные структуры

```
struct tnode {          /* the tree node: */
    char *word;             /* points to the text */
    int count;              /* number of occurrences */
    struct tnode *left;     /* left child */
    struct tnode *right;    /* right child */
};
```

Синонимы типов

```
typedef int Length;

Length len, maxlen;

typedef char *String;

String name;

typedef struct tnode *Treeptr;

typedef struct tnode {   /* the tree node: */
    char *word;              /* points to the text */
    int count;               /* number of occurrences */
    Treeptr left;            /* left child */
    Treeptr right;           /* right child */
} Treenode;
```

Ввод вывод

```c
/* reads single char from stdin
   return EOF constant on end of file */
int getchar()

/* print single char on stdout */
int putchar(int);

/* formatted output function */
int printf(char *format, ...)

int number = 42;
printf("number = %d", number);

/* formatted input function */
int scanf(char *format, ...)

int len;
scanf("%d", &len);
```

Форматный вывод

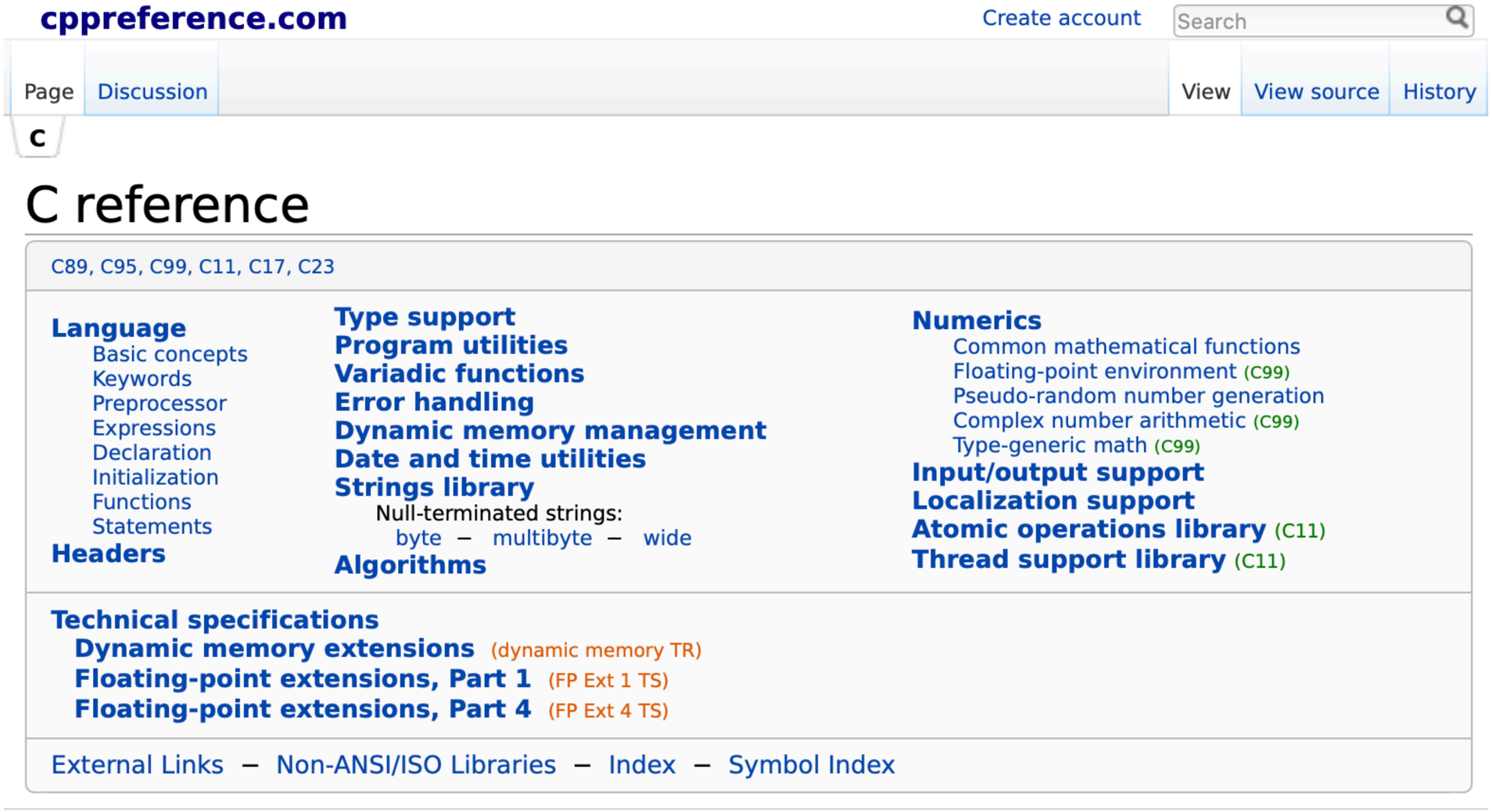
CHARACTER	ARGUMENT TYPE; PRINTED AS
`d, i`	`int`; decimal number.
`o`	`int`; unsigned octal number (without a leading zero).
`x, X`	`int`; unsigned hexadecimal number (without a leading `0x` or `0X`), using `abcdef` or `ABCDEF` for 10, ..., 15.
`u`	`int`; unsigned decimal number.
`c`	`int`; single character.
`s`	`char *`; print characters from the string until a `'\0'` or the number of characters given by the precision.
`f`	`double`; [−]*m*.*dddddd*, where the number of *d*'s is given by the precision (default 6).
`e, E`	`double`; [−]*m*.*ddddde*±*xx* or [−]*m*.*ddddd*`E`±*xx*, where the number of *d*'s is given by the precision (default 6).
`g, G`	`double`; use `%e` or `%E` if the exponent is less than −4 or greater than or equal to the precision; otherwise use `%f`. Trailing zeros and a trailing decimal point are not printed.
`p`	`void *`; pointer (implementation-dependent representation).
`%`	no argument is converted; print a `%`.

Стандартная библиотека

```
<assert.h>  <float.h>   <math.h>    <stdarg.h>  <stdlib.h>
<ctype.h>   <limits.h>  <setjmp.h>  <stddef.h>  <string.h>
<errno.h>   <locale.h>  <signal.h>  <stdio.h>   <time.h>
```

Справочное руководство

https://en.cppreference.com/w/c



Как изучить C?

- Выполнять задачи из книг
- Решать задачи на https://leetcode.com
- Переписать лабораторные работы с Pascal на C
- Написать свой проект на C. Например, лабораторный практику по ОС :)