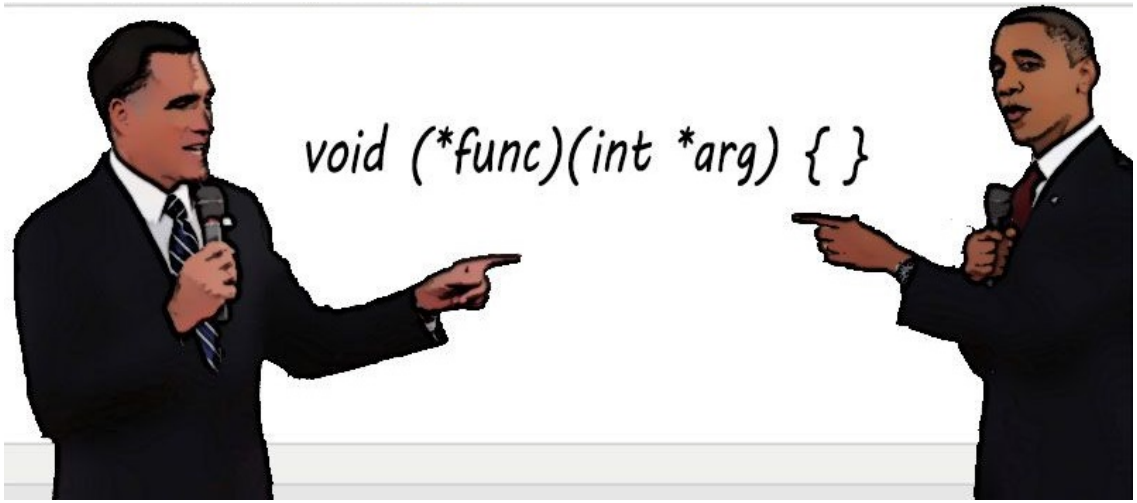
# Function Pointer and C Standard Library

魏恒峰

hfwei@nju.edu.cn

2017 年 11 月 24 日

# Function Pointer



# Sort

# Sort

## Compare & Swap

# Sort for any types

## Compare & Swap

# Sort for any types

## Compare & Swap

sort_ints

sort_floats

sort_strings

sort_persons

...

# Sort for any types

## Compare & Swap

sort_ints

sort_floats

sort_strings

sort_persons

...

(#include <stdlib.h>)

```
void qsort (void *base, size_t num, size_t size,
      int (*compar)(const void*, const void*));
```

(#include <stdlib.h>)

```
void qsort (void *base, size_t num, size_t size,
      int (*compar)(const void*, const void*));
```

(compare.c)

```c
int (*fptr)(int);  // fptr is a function pointer

int square(int num) {
  return num * num;
}
```

```c
int (*fptr)(int);  // fptr is a function pointer

int square(int num) {
  return num * num;
}
```

```c
int n = 5;
fptr = square;  // fptr points to a function
fptr(n);        // call ''square''
```

```c
int (*fptr)(int);  // fptr is a function pointer

int square(int num) {
  return num * num;
}
```

```c
int n = 5;
fptr = square;  // fptr points to a function
fptr(n);        // call ''square''
```

(fptr-square.c)

```
typedef int (*predicate)(int n);^^I// type!
```

```
typedef int (*predicate)(int n);^^I// type!
```

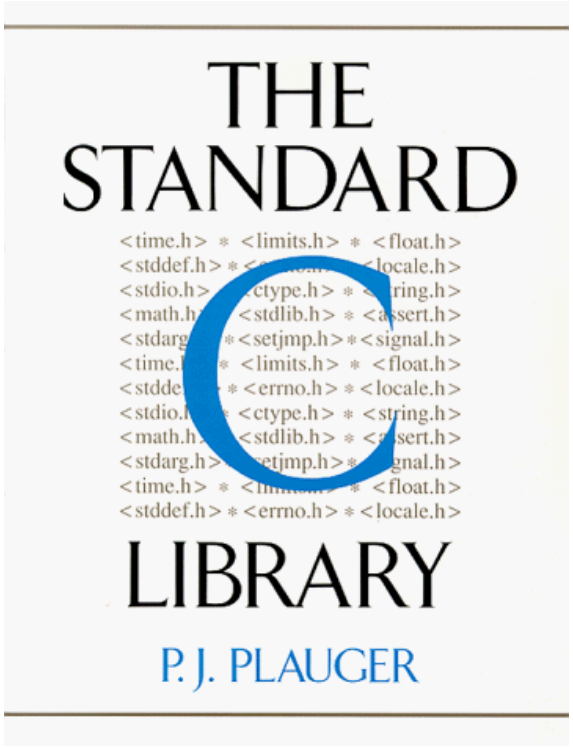

```
void filter(int *vals, int n, predicate cond);
```

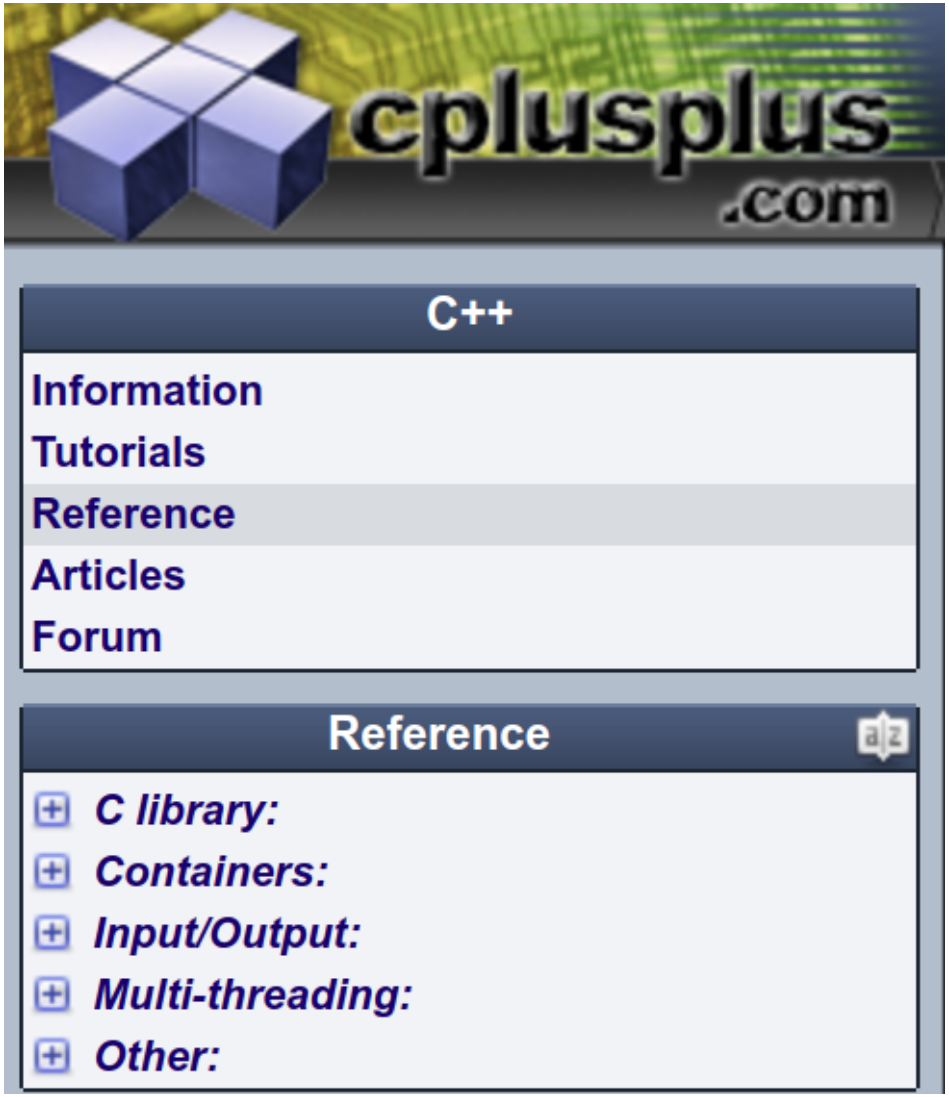
```
typedef int (*predicate)(int n);^^I// type!
```


```
void filter(int *vals, int n, predicate cond);
```

(filter-fp.c)

# C Standard Library




cplusplus
.com
C++
Information
Tutorials
Reference
Articles
Forum
Reference
C library:
Containers:
Input/Output:
Multi-threading:
Other:

- *C library:*
  - <cassert> (assert.h)
  - <cctype> (ctype.h)
  - <cerrno> (errno.h)
  - <cfenv> (fenv.h)
  - <cfloat> (float.h)
  - <cinttypes> (inttypes.h)
  - <ciso646> (iso646.h)
  - <climits> (limits.h)
  - <clocale> (locale.h)
  - <cmath> (math.h)
  - <csetjmp> (setjmp.h)
  - <csignal> (signal.h)
  - <cstdarg> (stdarg.h)
  - <cstdbool> (stdbool.h)
  - <cstddef> (stddef.h)
  - <cstdint> (stdint.h)
  - <cstdio> (stdio.h)
  - <cstdlib> (stdlib.h)
  - <cstring> (string.h)
  - <ctgmath> (tgmath.h)
  - <ctime> (time.h)
  - <cuchar> (uchar.h)
  - <cwchar> (wchar.h)
  - <cwctype> (wctype.h)

(#include <assert.h>)

```
scanf(''%d'', &n);
assert(n > 0)
```

(#include <assert.h>)

```
scanf(''%d'', &n);
assert(n > 0)
```

```
int *vals = malloc(sizeof(int) * n);
assert(vals != NULL)
```

(#include <ctype.h>)

```
isdigit
isalpha
isalnum

islower
isupper

isspace

tolower
toupper
```

(#include <limits.h>)

CHAR_MIN

CHAR_MAX

INT_MIN

INT_MAX

(#include <math.h>)

sin

cos

exp

log

pow

sqrt

ceil

floor

(#include <stdarg.h>)

```
int printf(const char *format, ...);

printf(''%d %c %s'', num, ch, str);
```

(1) unnamed

(2) # unknown

(#include <stdarg.h>)

```
int printf(const char *format, ...);

printf(''%d %c %s'', num, ch, str);
```

(1) unnamed

(2) # unknown

(3) "$\cdots$" must be at the end

(4) $\geq 1$ named argument

(#include <stdarg.h>)

```
int printf(const char *format, ...);

printf(''%d %c %s'', num, ch, str);
```

(1) unnamed

(2) # unknown

(3) "· · ·" must be at the end

(4) $\geq 1$ named argument

va_list // type

va_start

va_arg

va_end

(#include <stdarg.h>)

```
int printf(const char *format, ...);

printf(''%d %c %s'', num, ch, str);
```

(1) unnamed

(2) # unknown

(3) "· · ·" must be at the end

(4) $\geq 1$ named argument

va_list // type

va_start

va_arg

va_end

(miniprintf.c)

## (Command-line arguments)

```
int main(int argc, char *argv[]) {
}
```

argv[0]: program name

(Command-line arguments)

```
int main(int argc, char *argv[]) {
}
```

argv[0]: program name

(add-cmdarg.c)

(Command-line arguments)

```
int main(int argc, char *argv[]) {
}
```

argv[0]: program name

(add-cmdarg.c)

(man ls)

(#include <stddef.h>)

```
typedef /*implementation-defined*/ size_t;
```

(#include <stddef.h>)

```
typedef /*implementation-defined*/ size_t;
```

```
sizeof(int)

void* malloc (size_t size);
```

(#include <stdio.h>)

scanf

printf

getchar

putchar

fopen

fclose

EOF

(#include <stdlib.h>)

atoi

atof

srand

rand

malloc

free

bsearch

qsort

(#include <string.h>)

```
char book[] = ''The C Book'';
char *pbook = ''The C Book'';
```

(#include <string.h>)

```
char book[] = ''The C Book'';
char *pbook = ''The C Book'';
```

strncopy

strncat

strncmp

strlen

strchr

strrchr

strstr

strtok

(#include <string.h>)

```
char book[] = ''The C Book'';
char *pbook = ''The C Book'';
```

'\0'

strncopy

strncat

strncmp

strlen

strchr

strrchr

strstr

strtok

(#include <string.h>)

```
char book[] = ''The C Book'';
char *pbook = ''The C Book'';
```

'\0'

strncopy

strncat

strncmp

strlen

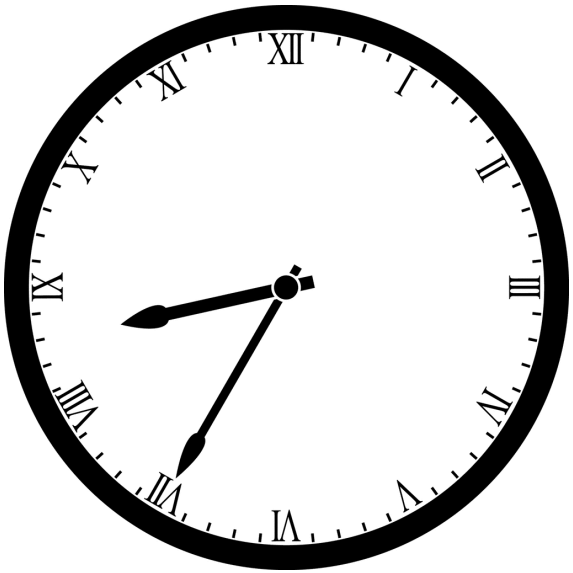
strchr

strrchr

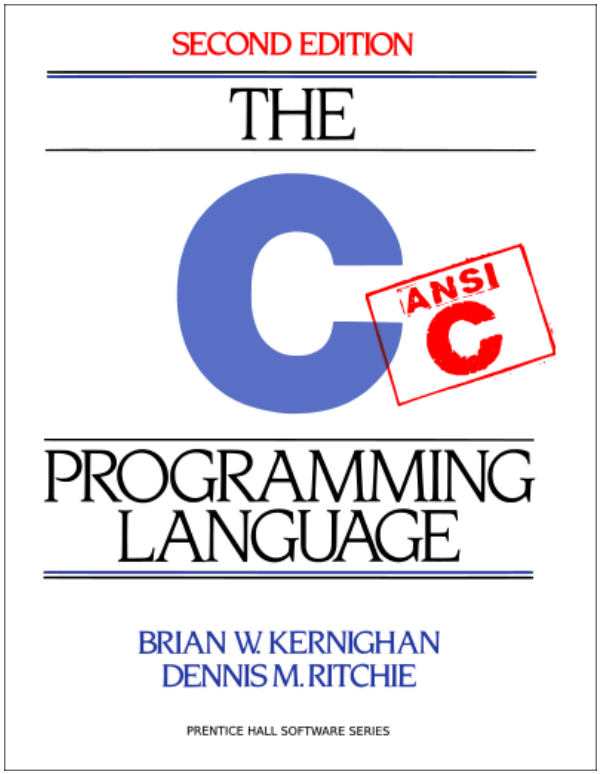
strstr

strtok

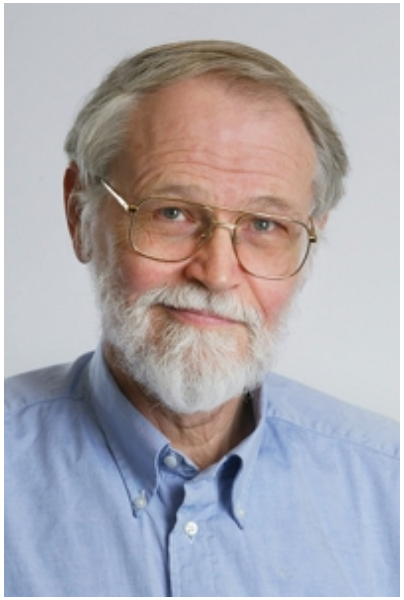
(#include <time.h>)

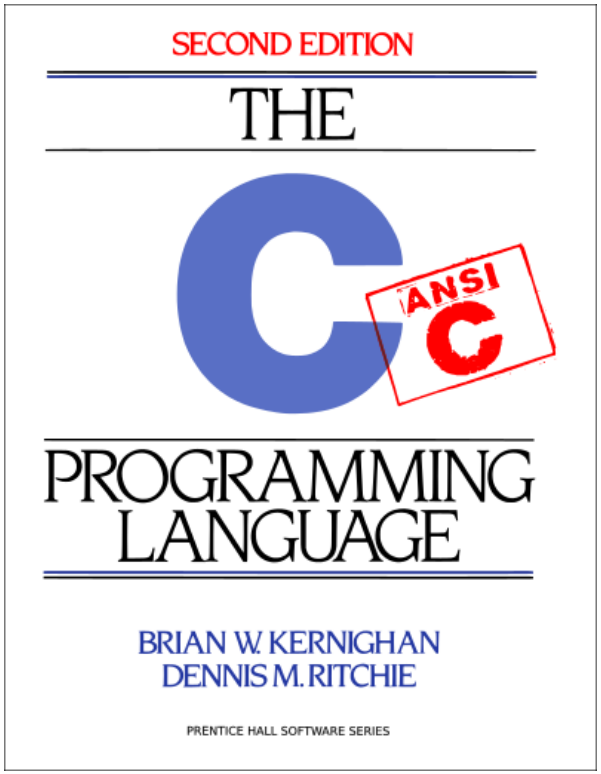


# Highly Recommended!



# Highly Recommended!







WHAT'S
NEXT?

# Thank You!