Programski jezik C

Tomaž Dobravec

C standard library

Del knjižnice libc: (več)

<assert.h>		Contains the assert macro, used to assist with detecting logical errors and other types of bug in debugging versions of a program.
<complex.h>	C99	A set of functions for manipulating complex numbers.
<inttypes.h>	C99	Defines exact-width integer types.
<limits.h>		Defines macro constants specifying the implementation-specific properties of the integer types.
<math.h>		Defines common mathematical functions.
<stdarg.h>		For accessing a varying number of arguments passed to functions.
<stdbool.h>	C99	Defines a boolean data type.
<stdint.h>	C99	Defines exact-width integer types.
<stdio.h>		Defines core input and output functions
<stdlib.h>		Defines numeric conversion functions, pseudo-random numbers generation functions, memory allocation, process control functions
<string.h>		Defines string-handling functions
<threads.h>	C11	Defines functions for managing multiple threads, mutexes and condition variables
<time.h>		Defines date- and time-handling functions

stdbool.h (C99)

```c
#include <stdbool.h>
#include <stdio.h>

int main() {
  bool iscem=true, nasel = false;

  if (nasel)
    printf("Nasel!\n");

  if (iscem)
    printf("Iscem!\n");

  printf("%d\n", iscem);
  printf("%d\n", nasel);
  printf("%d\n", iscem && nasel);
  printf("%d\n", iscem || nasel);
}
```

```
Iscem!
1
0
0
1
```

stdint.h (C99)

Dodatni celoštevilski tipi – z natančno določeno velikostjo

Specifier	Signing	Bits	Bytes
`int8_t`	Signed	8	1
`uint8_t`	Unsigned	8	1
`int16_t`	Signed	16	2
`uint16_t`	Unsigned	16	2
`int32_t`	Signed	32	4
`uint32_t`	Unsigned	32	4
`int64_t`	Signed	64	8
`uint64_t`	Unsigned	64	8

math.h

- Funkcije
  - `sin(), cos(), tanh(), asin(), acos(), …`
  - `exp(), log(), log10()`
  - `pow(), sqrt()`
  - `fabs()`
  - `trunc(), round(), ceil(), floor()`
  - …
- Ker se knjižnica `libm` ne poveže avtomatsko, je pri prevajanju potrebno dodati stikalo `-lm`

stdarg.h

```
#include<stdio.h>
#include<stdarg.h>

void printargument(int  num_args,  ...)
{
   va_list  arg_list;
   int  my_arg;

   va_start(arg_list,  num_args);

   //Print  until  zero
   for  (my_arg  = num_args;  my_arg  != 0; my_arg  = va_arg(arg_list,  int))
            printf("%d\n",  my_arg);

   va_end(arg_list);
}

int  main(void)
{
   printargument(5,10,15,0);
   return  0;
}
```

Naštevni tip - enum

- Prirejanje imen številskim konstantam
- Za lažjo berljivost programa

```
#include<stdio.h>

enum DAN{Ne, Po, To, Sr, Ce, Pe, So};

int main()  {
    enum DAN dan;
    dan = Po;
    printf("%d",dan);
    return 0;
}
```