

C 标准库 - <stdarg.h>

简介

`stdarg.h` 头文件定义了一个变量类型 `va_list` 和三个宏，这三个宏可用于在参数个数未知（即参数个数可变）时获取函数中的参数。

可变参数的函数通过在参数列表的末尾是使用省略号(...)定义的。

库变量

下面是头文件 `stdarg.h` 中定义的变量类型：

| 序号 | 变量 & 描述 |
|----|--|
| 1 | <code>va_list</code> 这是一个适用于 <code>va_start()</code> 、 <code>va_arg()</code> 和 <code>va_end()</code> 这三个宏存储信息的类型。 |

库宏

下面是头文件 `stdarg.h` 中定义的宏：

| 序号 | 宏 & 描述 |
|----|---|
| 1 | <code>void va_start(va_list ap, last_arg)</code> 这个宏初始化 <code>ap</code> 变量，它与 <code>va_arg</code> 和 <code>va_end</code> 宏是一起使用的。 <code>last_arg</code> 是最后一个传递给函数的已知的固定参数，即省略号之前的参数。 |
| 2 | <code>type va_arg(va_list ap, type)</code> 这个宏检索函数参数列表中类型为 <code>type</code> 的下一个参数。 |
| 3 | <code>void va_end(va_list ap)</code> 这个宏允许使用了 <code>va_start</code> 宏的带有可变参数的函数返回。如果在从函数返回之前没有调用 <code>va_end</code> ，则结果为未定义。 |



1 篇笔记



写笔记



变参使用实例 `test.c`：

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

```
void tLog(char *sInfo, ...);

int main(int argc, char *argv[])
{
    char str[4+1] = {0};
    strcpy(str, "INFO");
    tLog(str, argv[1], argv[2], argv[3]);
    return 0;
}

void tLog(char *sInfo, ...)
{
    char *str    = NULL;
    char *str1 = NULL;
    char *str2 = NULL;
    va_list sArgv;          // 申请参数列表变量
    va_start(sArgv, sInfo); // 申明最后一个传递给函数的已知的固定参数
    /* 依次读取固定参数 sInfo 后的 参数 */
    str  = va_arg(sArgv, char*);
    str1 = va_arg(sArgv, char*);
    str2 = va_arg(sArgv, char*);
    va_end(sArgv);
    printf("运行输出: [%s,%s%s%s]\n", sInfo, str, str1, str2);
}
```

编译 **gcc test.c** 得到 **a.out**。

运行程序：

```
$ ./a.out  Runoob Google Facebook
运行输出: [INFO,RunoobGoogleFacebook]
```

飞奔的蚂蚁2018 10个月前 (05-31)