

基于 Windows 的 μ C/OS-II 操作系统

实验指导书

本次实验将完成下列过程：

- (1) 在 Windows 平台下搭建 μ C/OS-II 开发环境。
- (2) 初始化 μ C/OS-II 内核，并完成 5 个任务的创建和启动。

本次实验基于以下软件和环境完成：

- (1) Windows11 (Windows XP 及以上即可)
- (2) DOSBox 0.74-3
- (3) Borland C++ 3.1
- (4) μ C/OS-II v2.52

μ C/OS-II 由 Micrium 公司提供，是一个可移植、可固化的、可裁剪的、占先式多任务实时内核，它是专门为计算机的嵌入式应用设计的，具有执行效率高、占用空间小、实时性能优良和可扩展性强等特点，适用于多种微处理器，微控制器和数字处理芯片。从 1992 年开始， μ C/OS-II 已经广泛使用在从照相机到航空电子产品的各种应用中。

一、在 Windows 平台下搭建 μ C/OS-II 开发环境

μ C/OS 最早发布于 1992 年，其在 Windows XP 以上系统运行会出现版本不兼容的问题，于是我们使用 DOSBox 来解决此问题。

DOSBox 是一款免费、开源的 MS-DOS 模拟器，可以为 DOS 程序提供执行环境，使这些程序可以正常运行于大多数现代计算机上的不同操作系统。

Borland C++ 是一款运行于 MS-DOS 系统下的 C++ 程序开发环境，此实验中，我们使用 Borland C++ 3.1 作为 μ C/OS-II 的集成开发环境 (IDE)。

1. 安装 DOSBox

(1) 创建项目文件夹

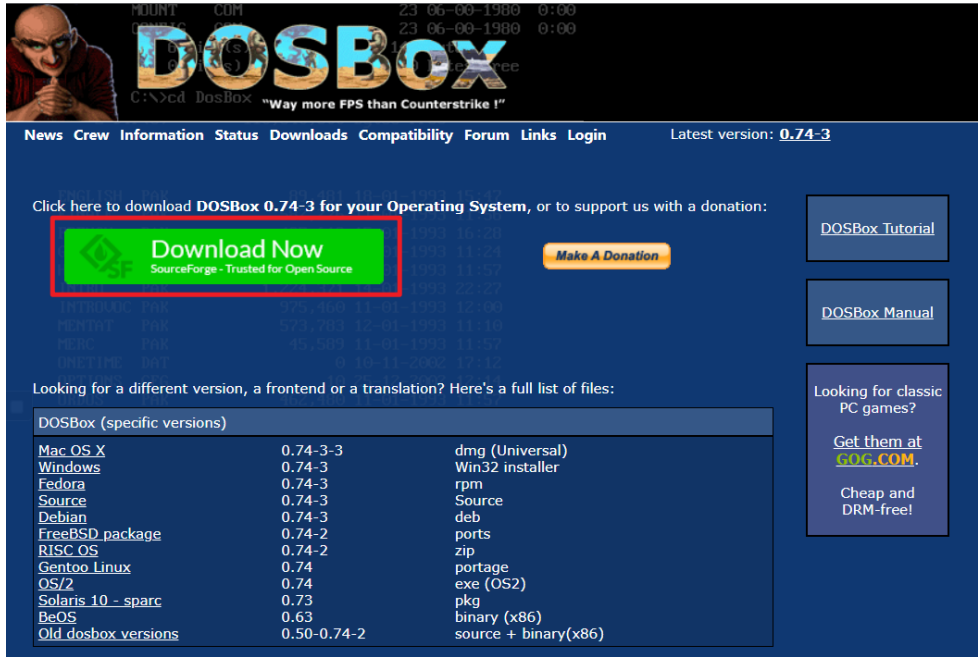
在 Windows D 盘根目录创建项目文件夹 uCOS-BC31。

在 uCOS-BC31 文件夹下创建 DOSBox-0.74-3、BORLANDC、uCOS-II、TEST-x86 四个文件夹。

(2) 安装 DOSBox

DOSBox 官网：<https://www.dosbox.com/>。

在官网下载 DOSBox 0.74-3，并将 DOSBox 安装至 uCOS-BC31 目录下 DOSBox-0.74-3 文件夹。



(3) 配置 DOSBox

进入 DOSBox-0.74-3 文件夹。

新建文件 DOSBox-BC31.conf，该文件为 DOSBox 配置文件。

双击运行 DOSBox 0.74-3 Options.bat，将打开的文件内容全选并复制到 DOSBox-BC31.conf 文件，并在最后添加以下内容：

（如果没有自动打开文件，请去 C:\Users\用户名\AppData\Local\DOSBox 下找到 dosbox-0.74-3.conf，打开该文件，全选并复制到 DOSBox-BC31.conf 文件。）

```
# 挂载上一级目录(uCOS-BC31)为 DOSBox C 盘所在位置
mount C ../.
# 设置 DOSBox 环境变量
set PATH=%PATH%;C:\BORLANDC\BIN
C:
# 进入工作目录，实际为 uCOS-BC31\TEST-x86
cd C:\TEST-x86
```

```

DOSBox-BC31.conf - 记事本
文件(F) 编辑(E) 格式(O) 视图(V) 帮助(H)
#           umb: Enable UMB support.
# keyboardlayout: Language code of the keyboard layout (or none).

xms=true
ems=true
umb=true
keyboardlayout=auto

[ipx]
# ipx: Enable ipx over UDP/IP emulation.

ipx=false

[autoexec]
# Lines in this section will be run at startup.
# You can put your MOUNT lines here.
# 挂载上一级目录(uCOS-BC31)为DOSBox C盘所在位置
mount C ../.
# 设置DOSBox环境变量
set PATH=%PATH%;C:\BORLANDC\BIN
C:
# 进入工作目录，实际为uCOS-BC31\TEST-x86
cd C:\TEST-x86

```

调整窗口大小:

默认 DOSBox 窗口比较小，我们可以通过配置文件修改，在 DOSBox-BC31.conf 文件内搜索 windowresolution。

将 windowresolution=original 改为 windowresolution=1366x800。

将 output=surface 改为 output=opengl。

```

#windowresolution=original
windowresolution=1366x800
#output=surface
output=opengl

```

返回 uCOS-BC31 目录，新建 RUN-DOSBox-BC31.bat 文件，添加以下内容：

```

CD DOSBox-0.74-3
DOSBox.exe -conf DOSBox-BC31.conf

```

RUN-DOSBox-BC31.bat 为 DOSBox 启动文件，双击即可启动 DOSBox。

补充：DOSBox 命令行

切换盘符命令：“X:”

显示文件、文件夹的命令（列表命令）：DIR

改变文件名的命令：REN

复制文件命令：COPY、XCOPY

删除命令：DEL、DELTREE

建立子目录（建立文件夹）的命令：MD

改变当前文件夹的命令：CD

删除目录的命令：RD

通配符：* 和 ?

退出：EXIT

2. 安装 Borland C++ 3.1

(1) 下载并安装 Borland C++ 3.1

我们可以在 [WinWorld](#) 网站上下载 Borland C++ 3.1 完整版本，但由于 Borland C 原版由 15 个软盘镜像组成，这里我们已经将其安装文件解压并打包在文件“Borland CPP 3.1 完整版.zip”下，可从文末提供的网盘链接下载。

下载“Borland C++ 3.1 完整版”，并解压至 uCOS-BC31 目录 BCPP31 文件夹。由于 INSTALL.EXE 为 16 位程序，不能直接在 Win11 运行，所以我们需要在 DOSBox 中安装 Borland C++ 3.1。

(2) 安装 Borland C++ 3.1

双击 RUN-DOSBox-BC31.bat，可以看到 DOSBox 成功启动。

在 DOS 下输入

```
cd ../BCPP31
```

```
INSTALL.EXE
```

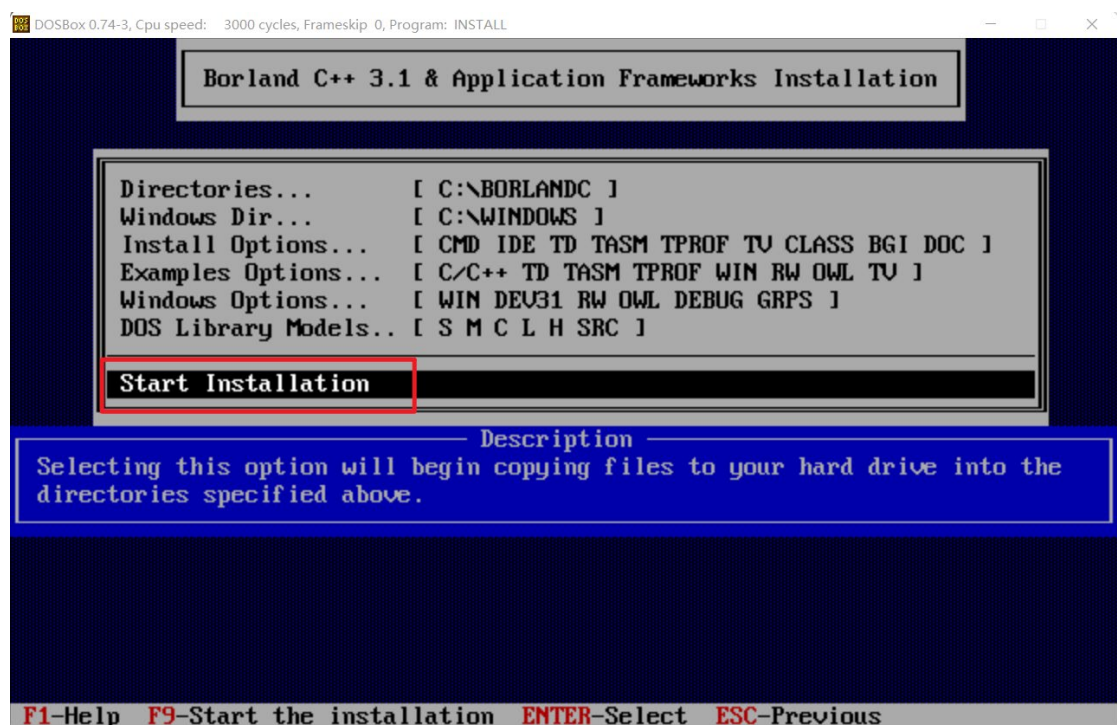
即可进入 Borland C++ 安装界面，按照下面设置安装即可：

Enter the SOURCE drive to use: C

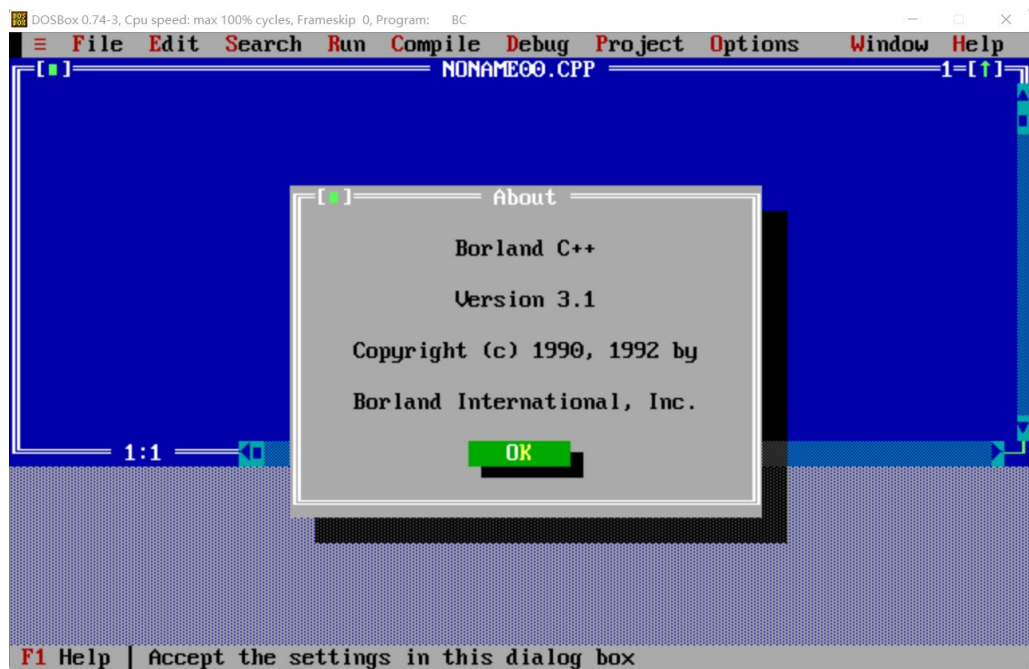
Enter the SOURCE Path: \BCPP31

Start Installation

遇到警告按 Enter 确认即可



安装完成后，删除 BCPP31 解压文件夹，在 DOS 下输入 bc 即可进入 Borland C++ 软件界面，也可以在 DOS 下用 bcc 命令来编译 C 程序。



3. 下载 μ C/OS-II 源码和实验程序

使用文末提供的网盘下载“uCOS-II v2.52 源码 zip”，并解压至 uCOS-BC31 目录下 uCOS-II 文件夹。其中，






SOURCE 目录存放的是 uCOS-II v2.52 内核源代码。

BLOCKS 目录存放的是 uCOS-II v2.52 与 PC 相关的函数。

Ix86L 目录存放的是在 80x86 处理器上运行 uC/OS-II 而必须的一些代码。

使用文末提供的网盘下载“TEST-x86.zip”，并解压至 uCOS-BC31 目录下 TEST-x86 文件夹，TEST-x86 目录存放的是实验所需的代码。

现在 uCOS-BC31 目录下有以下文件，检查一下是否一致。

 BORLANDC	2021/12/28 09:10	文件夹	
 DOSBox-0.74-3	2021/12/28 09:44	文件夹	
 TEST-x86	2021/12/28 01:12	文件夹	
 uCOS-II	2021/12/27 23:37	文件夹	
 RUN-DOSBox-BC31.bat	2021/12/28 01:07	Windows 批处理文件	1 KB

至此 μ C/OS-II 开发环境已经搭建完毕，下面就可以用 μ C/OS-II 内核代码开始实验了。

二、μC/OS-II 任务创建和启动实验

1. μC/OS-II 启动过程

在调用μC/OS-II 任何服务之前，μC/OS-II 要求首先调用初始化函数 OSInit(); 这个函数的目的就是在整个系统启动之前，初始化所有的变量和数据结构。调 OSInit 以后，任务控制块缓冲池中有 OS_MAX_TASK 个任务控制块，事件控制缓冲区中有 OS_MAX_EVENTS 个事件控制块，消息队列缓冲池 OS_Q 中有 OS_MAX_QS 个消息队列控制块。

```
1. int main(void)
2. {
3.     OSInit(); /* 系统初始化*/
4.     /* 创建主任务*/
5.     OSTaskCreate(MainTask, (void *)0, &MainTask_Stk[MainTask_StkSize-
        1], MainTask_Prio);
6.     OSStart(); /* 开始任务调度*/
7.     return 0;
8. }
```

2. 实验过程

(1) 实验要求

初始化μC/OS-II 内核，并完成 TASK #1 – TASK #5 的创建和启动。

(请参考 TEST-x86 目录下 EXP.pdf 中 29-40 页的 EXAMPLE #2，完成 TEST.C 文件；修改 TEST.MAK 文件，使其能够正确编译；执行编译产生的 TEST.EXE 文件。)

(2) 实验过程

TEST-x86/EXP-STY 目录下有 2 个文件夹，其中，

SOURCE 目录下

INCLUDES.H 为 EXAMPLE 的头文件。

OS_CFG.H 为 EXAMPLE 的配置文件。

TEST.LNK 为 链接文件，包含了链接的库和链接参数。

TEST 目录下

MAKETEST.BAT 为 编译 EXAMPL 的批处理文件。

TEST.MAK 为 编译 EXAMPLE 的 Makefile 文件。

CLEAN.BAT 为 清除运行过程文件的批处理文件。

复制 EXP-STY 文件夹为 EXP2 文件夹

编写 TEST.C 文件，并将 TEST.C 移动到 EXP2/ SOURCE 目录下

完善 EXP2/ TEST 目录下的 TEST.MAK 文件

启动 DOSBox
 cd EXP2/TEST
 MAKETEST.BAT
 TEST.EXE

即可看到 EXP2 运行结果，出现如下界面，恭喜，实验成功！

The screenshot shows the uC/OS-II, The Real-Time Kernel interface. At the top, it says "uC/OS-II, The Real-Time Kernel" and "Jean J. Labrosse". Below that, it says "EXAMPLE #2". A table displays task statistics:

Task	Total Stack	Free Stack	Used Stack	ExecTime (uS)
TaskStart():	624	178	446	202
TaskClk() :	1024	700	324	724
Task1() :	1024	662	362	686
Task2() :	1024	970	54	994
Task3() :	1024	448	576	472
Task4() :	1024	956	68	980
Task5() :	1024	944	80	968

Below the table, it shows "#Tasks : 9" and "CPU Usage: 11 %". At the bottom right, it shows "80387 FPU", "2021-12-28 11:45:11", and "V2.52".

附：μC/OS-II 实验用到的软件

链接：<https://pan.baidu.com/s/1WguRnxQlyJoDPHNqDrV8ww>

提取码：7hva

参考资料：

- [1]. <https://blog.csdn.net/overflyme/article/details/51434336>
- [2]. <https://blog.csdn.net/u011436427/article/details/81333259>
- [3]. <https://blog.csdn.net/dhauwd/article/details/79680268>
- [4]. https://blog.csdn.net/weixin_43808473/article/details/106457668
- [5]. https://blog.csdn.net/weixin_34198762/article/details/86037220
- [6]. http://www.wernerzimmermann.name/media/files/uCOS-II_WIN32.htm