



JIGSAW OS 开发文档

KVZM Studios

修改日期：2022 年 12 月 6 日，适用于 JigsawOS 内部版本 1。

目录

JigsawOS 开发文档.....	1
概述.....	1
开发环境.....	1
开始之前.....	1
创建你的第一个 JigsawOS 应用程序.....	1
API 参考.....	错误!未定义书签。
API 解释与演示	3

JigsawOS 开发文档

概述

[JigsawOS](#) 基于 HariboteOS 开发，并在原基础上增添更多内容。关于 HariboteOS 的更多信息，请前往：[HariboteOS/index \(osask.jp\)](https://hariboteos.github.io/index(osask.jp)) 了解更多。

JigsawOS 应用程序使用 C 语言编写。

开发环境

如果你使用的是 Windows，请下载 JCode 对你的程序进行编辑与生成。

有关 JCode，请前往：<https://pigeon-ming.github.io/JCode/> 了解更多。

开始之前……

请确保你有足够的计算机基础知识储备，且具有一定的 C 语言编程能力，以便后续开发能够顺利进行。

有关 JigsawOS 的安装与运行，请参照：<https://kvzmstudios.github.io/JigsawOS/>
JigsawOS 正处于起步阶段，若在开发过程中遇到不便请谅解！

若在使用中遇到问题，请反馈至：kvzmstudios@outlook.com

创建你的第一个 JigsawOS 应用程序

步骤 1：创建并运行 hello.hrb

1. 启动 JCode。
2. 在“文件”菜单中点击“新建”。
3. 在文本框内输入以下代码：

```
#include "apilib.h"

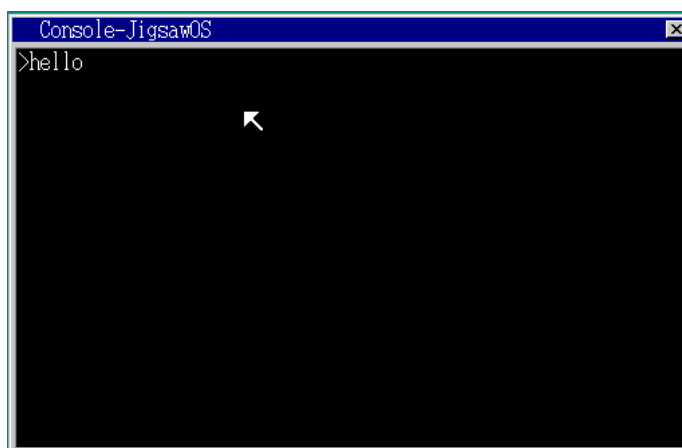
void HariMain(void)
{
    char *buf;
    int win;

    api_initmalloc();
    buf = api_malloc(250 * 150);
    win = api_openwin(buf, 250, 150, -1, "HelloWindow");
    api_boxfilwin(win, 8, 36, 141, 43, 6); /*浅蓝色*/
    api_putstrwin(win, 28, 28, 0, 12, "hello, JigsawOS");/*黑色*/
    for (;;) {
        if (api_getkey(1) == 0x0a) {
            break; /*按下回车键则 break; */
        }
    }
    api_end();
}
```

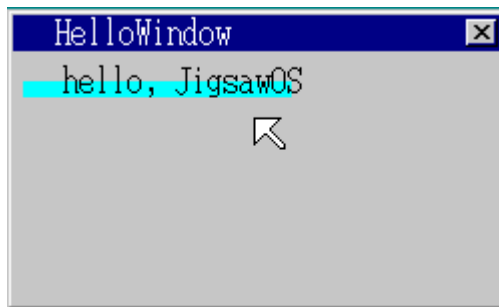
4. 在“文件”菜单中点击“另存为”，将文件命名为“**hello.c**”，然后点击保存。
(文件名必须为英文字符)
5. 点击工具栏中的“生成”，等待片刻。
6. 打开刚才另存为文件的位置，该位置会新增一个名为“**hello.hrb**”的文件。

Name	Date modified	Type	Size
hello.c	2022/12/4 21:03	C 源文件	1 KB
hello.hrb	2022/12/4 20:59	HRB File	1 KB

7. 利用 UltraISO 等工具将“**hello.hrb**”文件存入 JigsawOS 系统镜像: “**jigsawos.img**”中。
8. 启动 JigsawOS。
9. 在 JigsawOS 的命令行窗口中输入 hello。



10. 屏幕上将会出现你的应用，效果如下所示：



步骤 2：我们刚才做了什么？

我们使用了一些 JigsawOS API 创建了一个字符显示窗口。

运行该程序时，屏幕上将出现一个包含“hello, JigsawOS”和一条蓝色横线的窗口。

API 解释与演示

以下解释可能会伴随着 JigsawOS 开发而过时（解释可能未更新），若在使用中遇到问题，请反馈至：kvzmstudios@outlook.com

`void api_putchar(int c);`

概述：在命令行窗口内输出一个由参数“c”（Ascii 码）指定的字符。

示例：`api_putchar('p');`

在命令行内输出:p

`void api_putstr0(char *s);`

概述：在命令行窗口内输出字符串。

示例：`void api_putstr0("Hello,JigsawOS");`

在命令行内输出:Hello,JigsawOS

`void api_putstr1(char *s, int l);`

概述：在命令行窗口内输出长度为 l 的字符串。

示例：`void api_putstr1("Hello,JigsawOS", 14);`

在命令行内输出:Hello,JigsawOS

`void api_end(void);`

概述：终止应用程序。

`int api_openwin(char *buf, int xsiz, int ysiz, int col_inv, char *title);`

`void api_putstrwin(int win, int x, int y, int col, int len, char *str);`

`void api_boxfilwin(int win, int x0, int y0, int x1, int y1, int col);`

`void api_initmalloc(void);`

```
char *api_malloc(int size);

void api_free(char *addr, int size);

void api_point(int win, int x, int y, int col);

void api_refreshwin(int win, int x0, int y0, int x1, int y1);

void api_linewin(int win, int x0, int y0, int x1, int y1, int col);

void api_closewin(int win);

int api_getkey(int mode);

int api_alloctimer(void);

void api_inittimer(int timer, int data);

void api_settimer(int timer, int time);

void api_freetimer(int timer);

void api_beep(int tone);

int api_fopen(char *fname);

void api_fclose(int fhandle);

void api_fseek(int fhandle, int offset, int mode);

int api_fsize(int fhandle, int mode);

int api_fread(char *buf, int maxsize, int fhandle);

int api_cmdline(char *buf, int maxsize);

int api_getlang(void);
```

以下为 JigsawOS api:

```
int api_getmemorymessage(void);
```

概述: 取得目前空闲的内存大小, 单位: B。

```
int api_getjigsawversion(void);
```

概述：取得 JigsawOS 版本号。

```
int api_getjigsawBuildversion(void);
```

概述：取得 JigsawOS 内部版本号。