

项目申请书



项目 ID: 210010031

开发 iso 刻录 DDE 桌面应用

申请人: *Hollow Man*

最后修改时间: 2021.5.19

目录

1. 项目详细方案	1
2. 时间安排	4
A. 第 1 周 (7.1 - 5, 2021).....	4
B. 第 2-6 周 (7.6 - 8.9, 2021) 共计 5 周	4
C. 第 7 周 (8.10 - 15, 2021).....	4
D. 中期报告 (8.15, 2021).....	4
E. 第 8-13 周 (8.17 - 9.24, 2021) 共计 5 周半	4
F. 第 13-14 周 (9.25 – 9.30, 2021).....	4
G. 结项(9.30, 2021).....	5
H. 未来 (10.1, 2021 -)	5

1. 项目详细方案

方案一：使用 dd 命令实现 ISO 的制作

Linux dd 命令用于读取、转换并输出数据。

dd 可从标准输入或文件中读取数据，根据指定的格式来转换数据，再输出到文件、设备或标准输出。

参数说明：

if=文件名：输入文件名，默认为标准输入。即指定源文件。

of=文件名：输出文件名，默认为标准输出。即指定目的文件。

ibs=bytes：一次读入 bytes 个字节，即指定一个块大小为 bytes 个字节。

obs=bytes：一次输出 bytes 个字节，即指定一个块大小为 bytes 个字节。

bs=bytes：同时设置读入/输出的块大小为 bytes 个字节。

cbs=bytes：一次转换 bytes 个字节，即指定转换缓冲区大小。

skip=blocks：从输入文件开头跳过 blocks 个块后再开始复制。

seek=blocks：从输出文件开头跳过 blocks 个块后再开始复制。

count=blocks：仅拷贝 blocks 个块，块大小等于 ibs 指定的字节数。

conv=<关键字>，关键字可以有以下 11 种：

conversion：用指定的参数转换文件。

ascii：转换 ebcdic 为 ascii

ebcdic：转换 ascii 为 ebcdic

ibm：转换 ascii 为 alternate ebcdic

block：把每一行转换为长度为 cbs，不足部分用空格填充

unblock：使每一行的长度都为 cbs，不足部分用空格填充

lcase：把大写字符转换为小写字符

ucase：把小写字符转换为大写字符

swap：交换输入的每对字节

noerror：出错时不停止

notrunc：不截短输出文件

sync：将每个输入块填充到 ibs 个字节，不足部分用空（NUL）字符补齐。

--help: 显示帮助信息

--version: 显示版本信息

实例

在 Linux 下制作启动盘，可使用如下命令：

```
dd if=boot.img of=/dev/fd0 bs=1440k
```

将 testfile 文件中的所有英文字母转换为大写，然后转成为 testfile_1 文件，在命令提示符中使用如下命令：

```
dd if=testfile_2 of=testfile_1 conv=ucase
```

其中 testfile_2 的内容为：

```
$ cat testfile_2 #testfile_2 的内容
```

```
HELLO LINUX!
```

```
Linux is a free unix-type operating system.
```

```
This is a linux testfile!
```

```
Linux test
```

转换完成后，testfile_1 的内容如下：

```
$ dd if=testfile_2 of=testfile_1 conv=ucase #使用 dd 命令，大小写转换记录了 0+1 的读入  
记录了 0+1 的写出
```

```
95 字节（95 B）已复制，0.000131446 秒，723 KB/s
```

```
cmd@hdd-desktop:~$ cat testfile_1 #查看转换后的 testfile_1 文件内容
```

```
HELLO LINUX!
```

```
LINUX IS A FREE UNIX-TYPE OPERATING SYSTEM.
```

```
THIS IS A LINUX TESTFILE!
```

```
LINUX TEST #testfile_2 中的所有字符都变成了大写字母
```

由标准输入设备读入字符串，并将字符串转换成大写后，再输出到标准输出设备，使用的命令为：

```
dd conv=ucase
```

输入以上命令后按回车键，输入字符串，再按回车键，按组合键 Ctrl+D 退出，出现以下结果：

```
$ dd conv=ucase
```

```
Hello Linux! #输入字符串后按回车键
```

HELLO LINUX! #按组合键 Ctrl+D 退出，转换成大写结果

记录了 0+1 的读入

记录了 0+1 的写出

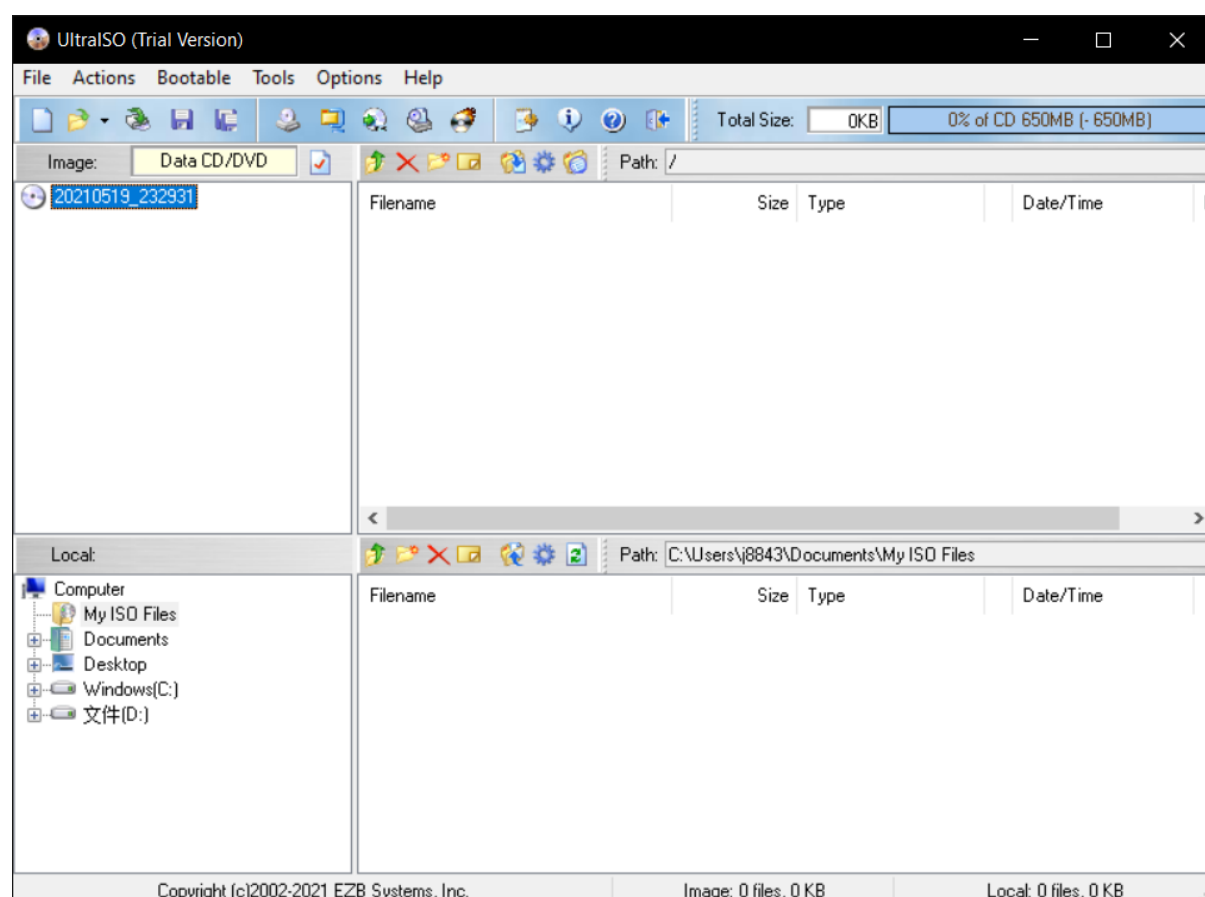
13 字节（13 B）已复制，12.1558 秒，0.0 KB/s

方案二：模仿 ventoy 原理进行实现

Ventoy 是一个创建可启动 U 盘的工具。利用 Ventoy 你只需把 ISO 文件拷贝到 U 盘中即可启动它。U 盘中可以同时存放多个、多种不同类型操作系统的 ISO 镜像，在启动时 Ventoy 会提供一个菜单来进行选择。同时支持 Legacy BIOS 和 UEFI 模式，支持 Windows、Linux、VMware、Xen 等常见的操作系统类型，对超过 100 个 ISO 文件进行了测试。

我也可以借助学习 ventoy 的实现原理进行实现。

编程计划



计划使用 python 语言进行编写，调用 Linux 中自带的 dd 命令进行 ISO 光盘的刻录，学习 Ventoy 的原理进行编写。

2. 时间安排

因为我今年的谷歌编程之夏已经完成，学校的期末考试也将于 6 月底结束，暑期也没有其它的安排。所以在 7 月份到 9 月份这 3 个月我可以全职工作（不限时间，每周至少 30 小时）。

我知道遇到问题时优先应该在互联网上搜索并参考有关资料，因而我只会和导师讨论自己无法克服的困难。我会每周通过电子邮件向导师发送一份“周期总结”，告知我到目前为止的工作进展和这周取得的进展。

以下是时间表：

A. 第 1 周 (7.1 - 5, 2021)

- 进一步了解 Ventory，查阅相关资料，并且熟悉代码库和其风格。

B. 第 2-6 周 (7.6 - 8.9, 2021) 共计 5 周

- 仿照 UltraISO，使用 QTDesigner 初步设计好图形界面。
- 进一步详细学习 Ventory 的原理。
- 如果在此期间时间有剩余，可以帮助社区解决一些其它 Issue，贡献社区。

C. 第 7 周 (8.10 - 15, 2021)

- 准备中期报告。
- 为可能的拖延预留的赶进度时间。

D. 中期报告 (8.15, 2021)

交付成果

- 使用 QTDesigner 初步设计好的图形界面。
- 帮助社区解决一些其它 Issue，贡献社区。

E. 第 8-13 周 (8.17 - 9.24, 2021) 共计 5 周半

- 进行图形界面的逻辑编写，实现刻录 ISO 功能。

F. 第 13-14 周 (9.25 - 9.30, 2021)

- 准备结项报告。
- 为可能的拖延预留的赶进度时间。

G. 结项(9.30, 2021)

交付成果

- 类似 UltraISO 软碟通的 openEuler DDE 桌面应用。可以完成 iso 刻录至 u 盘。

H. 未来 (10.1, 2021 -)

在成功完成此暑期 2021 项目后，我将继续为 openEuler 做出贡献并尽我所能继续维护。如果可能的话，我会很高兴在未来的暑期开源活动中成为 openEuler 社区的导师。