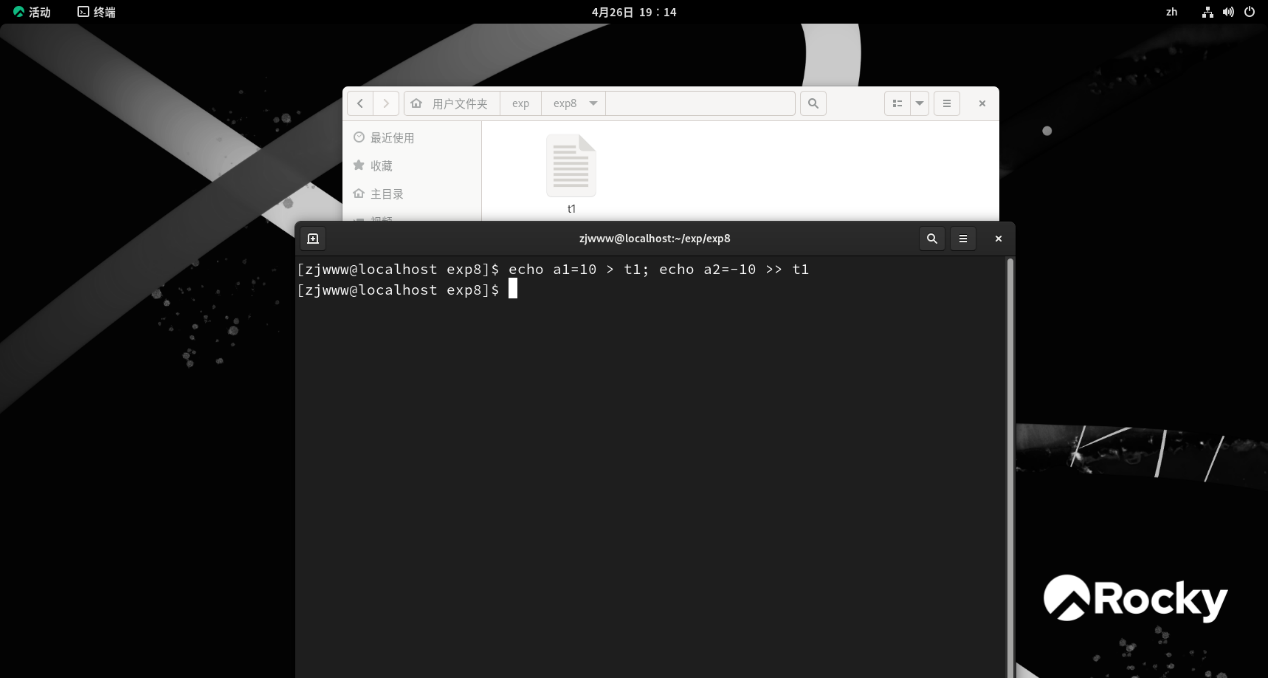
**实验八**

1. **命令解释与文件创建**
2. 命令解释

执行echo a1=10 > t1; echo a2=-10 >> t1



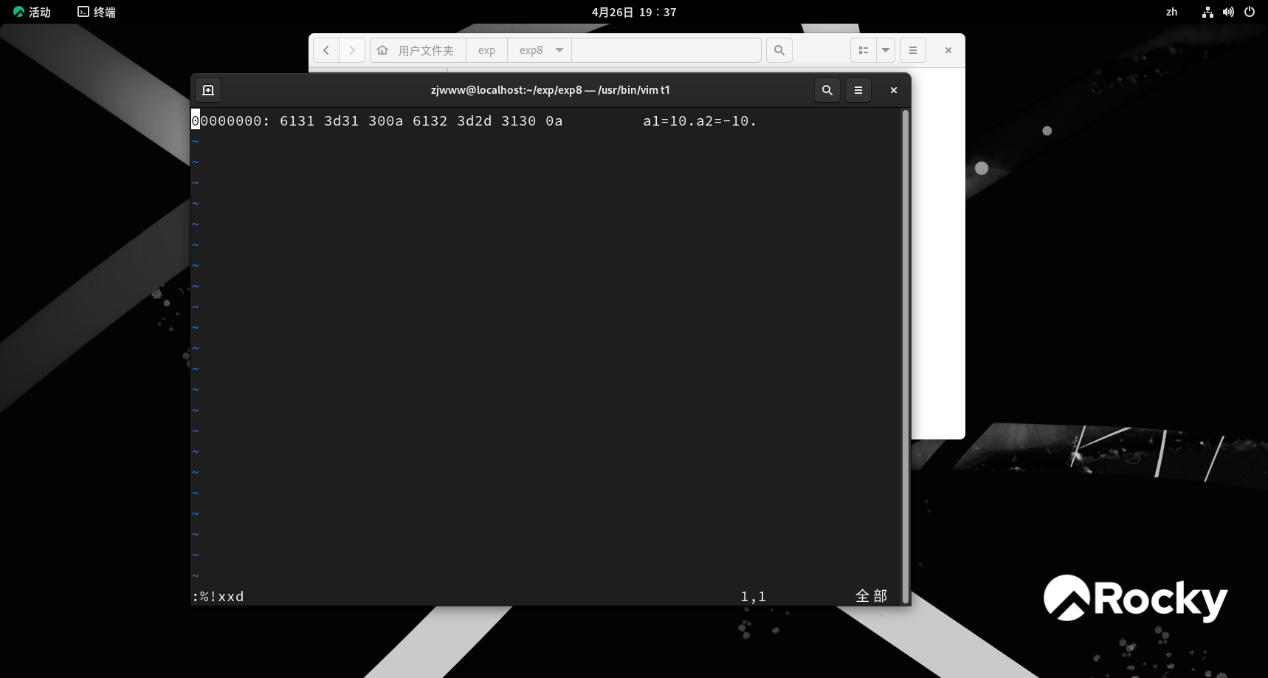
图形用户界面, 文本

AI 生成的内容可能不正确。

* echo a1=10 > t1：将字符串"a1=10"输出并重定向到文件t1（创建或覆盖）
* echo a2=-10 >> t1：将字符串"a2=-10"追加到文件t1末尾

1. 16进制查看

使用vi打开t1，在命令模式下输入:%!xxd，显示如下：



1. 16进制解释：

* 61 = 'a'；31 = '1'；3d = '='；31 = '1'
* 30 = '0'；0a = 换行符'\n'；61 = 'a'；32 = '2'
* 3d = '='；2d = '-'；31 = '1'；30 = '0'

1. 文件大小是13字节，因为：

* 第一行 a1=10\n 占6字节（5字符 + 1换行符）。
* 第二行 a2=-10\n 占7字节（6字符 + 1换行符）。

两行合计：6 + 7 = 13字节。

1. **文本文件生成程序**
2. 创建程序

创建并编写如下程序text\_gen.c

图形用户界面

AI 生成的内容可能不正确。

1. 编译运行

文本

AI 生成的内容可能不正确。

1. 16进制查看

图形用户界面

AI 生成的内容可能不正确。

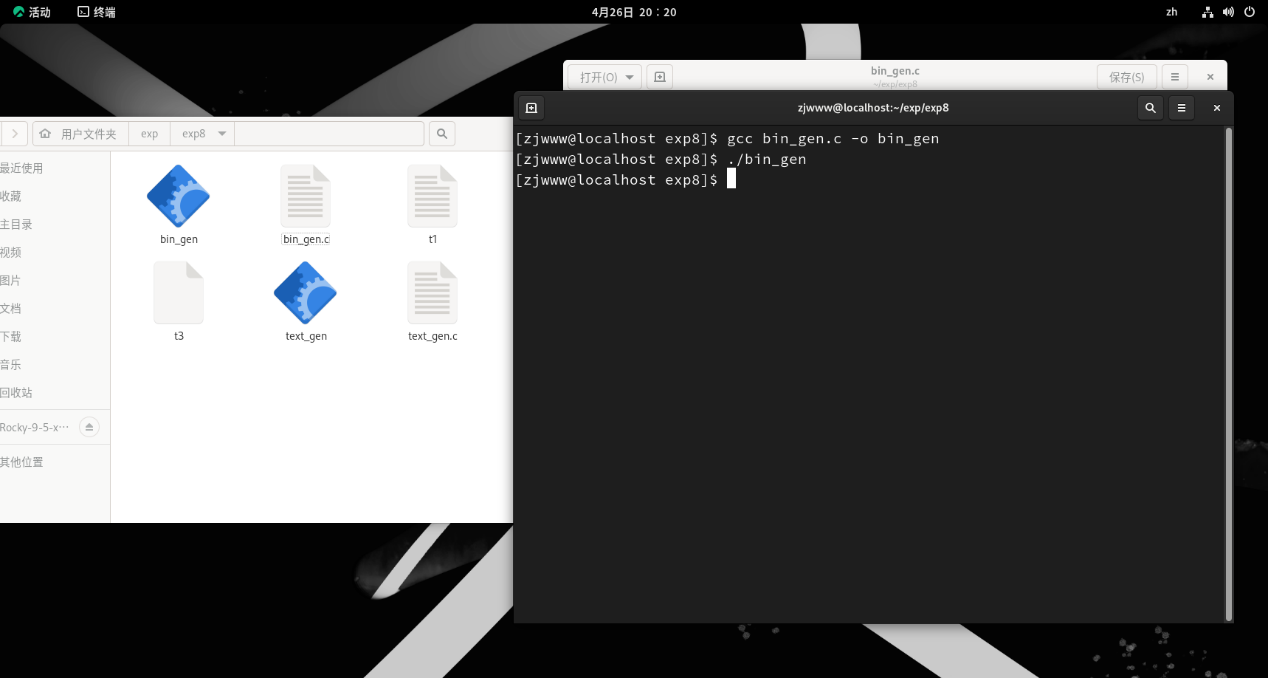
可以看到与t1的16进制查看完全相同

1. **二进制文件生成程序**
2. 创建并编写程序bin\_gen.c

图形用户界面

AI 生成的内容可能不正确。

1. 编译运行



1. 使用16进制查看

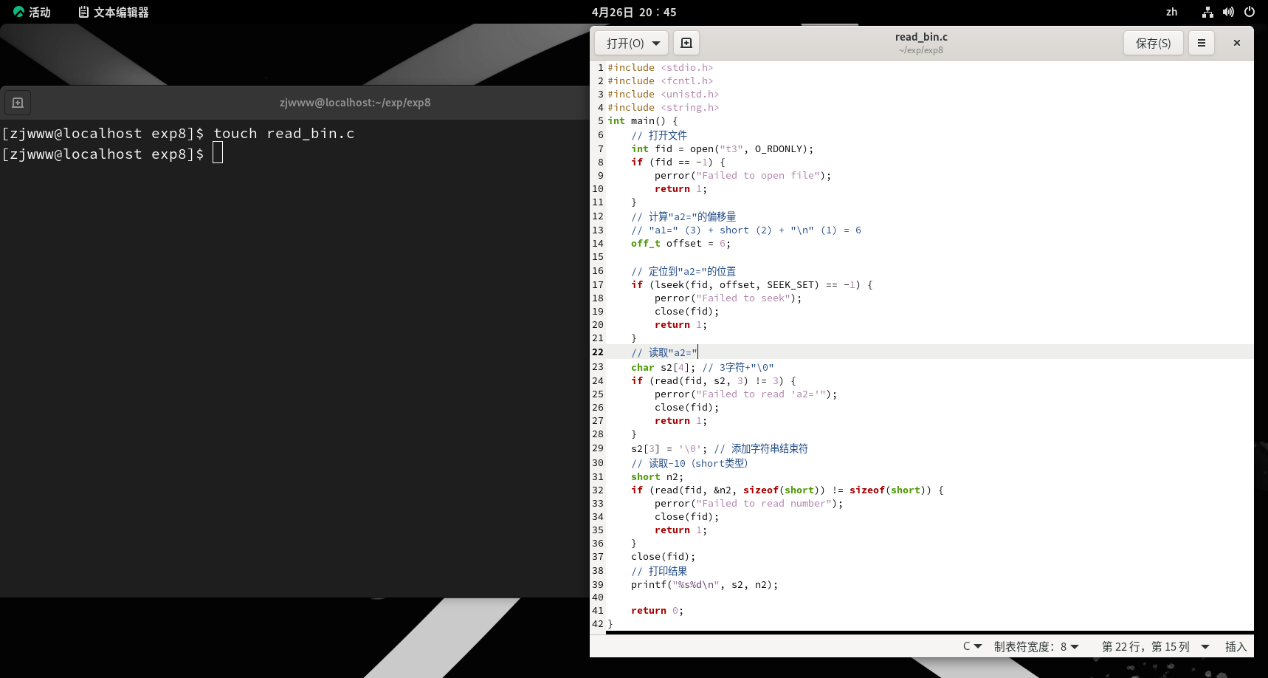
vi命令模式中执行%!xxd

电脑萤幕的截图

AI 生成的内容可能不正确。

可以看到在16进制下查看t3和t2的内容不一样，其中红色区域为10的补码，蓝色区域为-10的补码

1. **按数据类型读取文件**
2. 创建并编写读取程序



1. 编译运行

