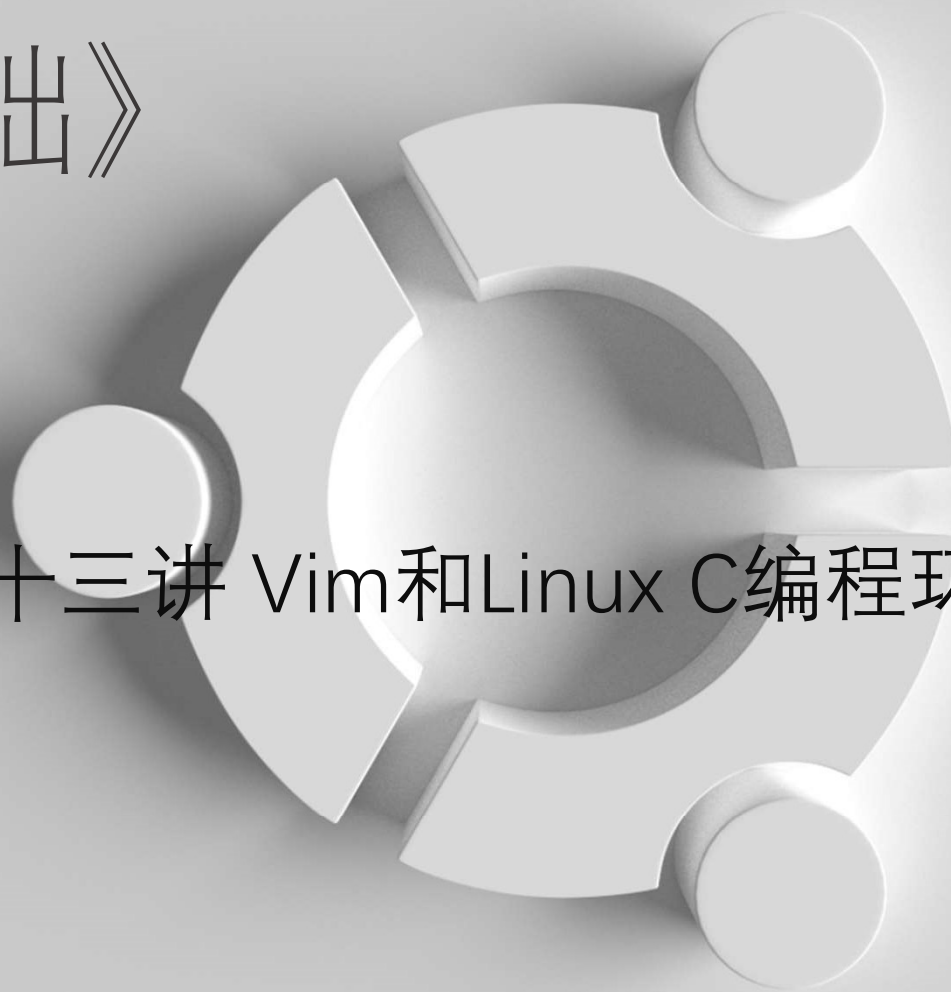


# 《Linux基础》

## 第十三讲 Vim和Linux C编程环境



## vim配置文件

---

- 安装vim后，配置文件：/etc/vim/vimrc，vim启动时会读取此文件。除此之外，在用户主目录下如果存在.vimrc，也会读取此文件，如果存在相同的配置，则.vimrc文件配置会覆盖/etc/vim/vimrc的配置。
- /usr/share/vim/vim74/colors是vim的配色主题文件所在目录，在用户主目录下的.vim/colors也可以保存主题文件，此目录通常需要自己创建。
- 通用的配置可以写在/etc/vim/vimrc，个性化设置可以放在~/.vimrc
- vim配置文件使用" 标记注释。

## 基本配置

---

set mouse=a “ 启用鼠标支持，a表示所有模式都支持

set t\_Co=256 “ 256颜色支持

set number “ 显示行号

set backspace=2 “ 设置退格键删除

set fileencodings=utf-8 “设置写入文件编码为utf-8

set encoding=utf-8 “ 设置vim缓冲区编码utf-8

set termencoding=utf-8 ” 设置终端显示编码为utf-8

set autoindent “ 自动缩进

set shiftwidth=4 “缩进4空格宽度

set tabstop=4 “ tab缩进4空格宽度

set expandtab “ tab转换为空格

## 颜色主题配置

---

- 设置暗色背景和暗色主题  
set background=dark  
colorscheme evening
- 设置亮色主题和亮色背景  
set background=light  
colorscheme tomorrow
- 两种模式选一种即可，colorscheme后面的是主题文件名称，不需要加上.vim。要确保存在主题文件，否则会出错。

# gcc基本介绍与安装

---

- gcc不是一个软件，全名是GNU Compiler Collection，是一个编译器集合。
- gcc支持C，C++，Objective-C，Fortran，Ada，Go语言的编译。
- gcc是GNU的一个项目，发布于GPL开源协议。
- UbuntuServer版本都自带gcc，如果没有则运行：  
`sudo apt update ; sudo apt install gcc。`
- 官网: <https://gcc.gnu.org/>

# gcc基本使用

---

- 使用vim test.c创建文件并编写一段C程序。
- gcc test.c即可编译，编译后的文件是a.out。
- ./a.out运行编译的程序。
- gcc编译后的默认名称是a.out
- gcc -o test test.c：编译后的文件是test，-o参数后面跟输出文件的名称。

# ELF文件

---

- gcc test.c编译后的输出文件是ELF格式的。
- 而扩展名和文件类型并没有关系，Windows是设计必须使用扩展名和对应的运行程序去关联。Windows上采用的PE格式作为可执行文件格式，.exe是Windows上的可执行文件的扩展名。
- 在Linux上采用的ELF格式不必在文件名中体现出来。但是如果使用vim打开编译后的文件，在最开始会看到ELF三个字符表明是ELF文件。
- readelf -h ./test可以查看test程序的ELF文件头信息。

# glibc

---

- glibc是Linux上标准C的函数库实现。
- .so文件是Linux上的共享库文件，相当于Windows上的.dll文件。
- glibc的实现库文件是/lib/x86\_64-linux-gnu/libc.so.6链接到/lib/x86\_64-linux-gnu/libc-\*\*\*.so; 星号表示版本，比如，在UbuntuServer16.04上是2.23，UbuntuServer18.04上是2.27。
- gcc在编译C语言代码的时候要用到glibc。



## main函数的参数和返回值

---

```
int main (int argc, char* argv[]) {  
    /*  
        .....  
        ..... //something code  
    */  
    return 0;  
}
```

- main函数的返回值是程序的退出状态表示程序是否正确执行。
- argc是传递给程序参数的个数，argv是每个参数的值，argv[0]永远都是程序文件的名称。但是这个名称是带有路径信息的。

# 支持参数的程序

---

- 输出传递给程序的参数:

```
#include <stdio.h>
int main(int argc, char* argv[])
{
    printf("program name: %s\n", argv[0]);
    for (int i=1; i<argc; i++)
        printf("%s ", argv[i]);
    if (argc>1)
        printf("\n");
    return 0;
}
```

# 编程训练

---

- 编写一个对整数排序的程序，通过参数获取数字，并输出排序结果。
- 使用插入排序即可，插入排序示例(nms是数字数组)：

```
int k, j, tmp;
for (k=1; k<N; k++) {
    tmp = nms[k];
    for(j=k; j>0 && nms[j-1]>tmp; j--)
        nms[j] = nms[j-1];
    nms[j] = tmp;
}
```

- 提示：

通过atoi(argv[i])可以把输入的参数转换成整数。

int \*n = (int\*)malloc(sizeof(int)\*10);会申请一个长度为10\*sizeof(int)字节的数组，并返回一个int\*类型的指针。使用free(n)释放掉申请的内存。