**嵌入式程序设计大作业**

### (1)第二章 Shell编程

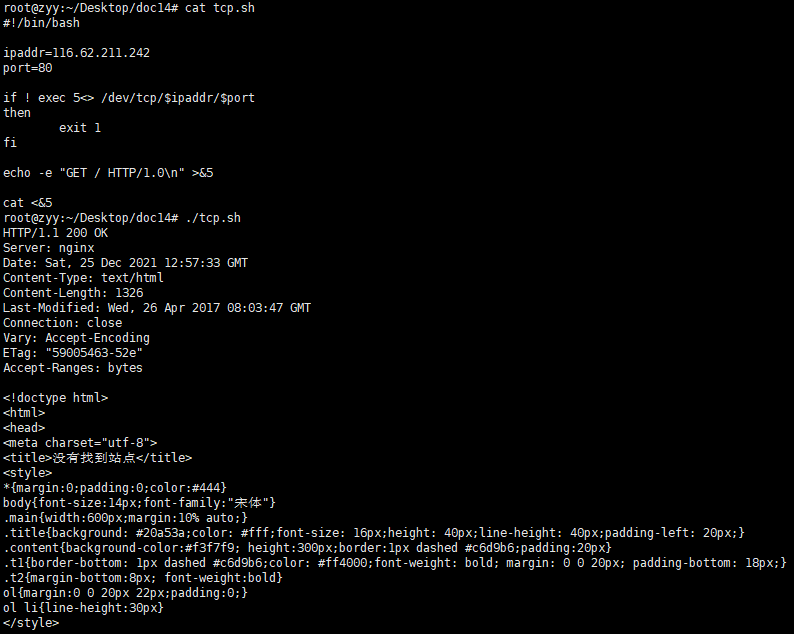
shell脚本的相关使用：检擦http协议的80端口是否打开。

dev/tcp & /dev/udp：这两个神奇的目录为bash编程提供了一种可以进行网络编程的功能。在bash程序中使用/dev/tcp/ip/port的方式就可以创建一个scoket作为客户端去连接服务端的ip:port。我们用一个检查http协议的80端口是否打开的例子来说明它的使用方法：

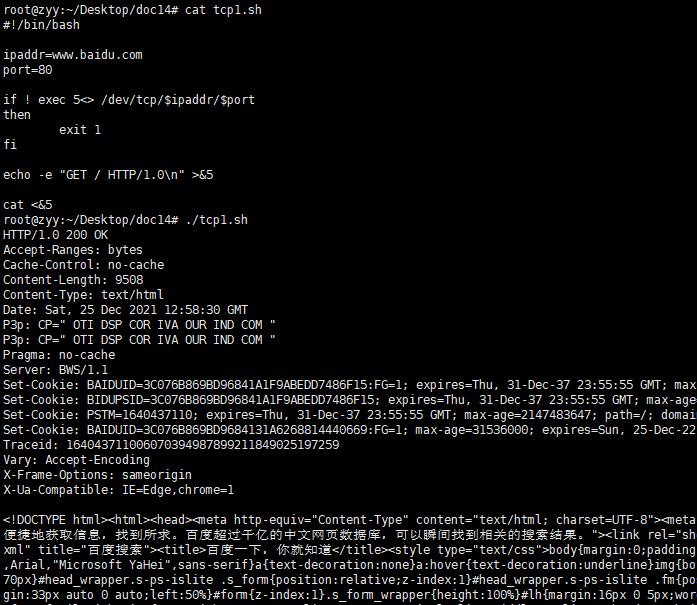
①未打开80端口：

# vi tcp.sh

# chmod 755 tcp.sh



②打开80端口



### (2)第三章 Linux系统C语言开发工具

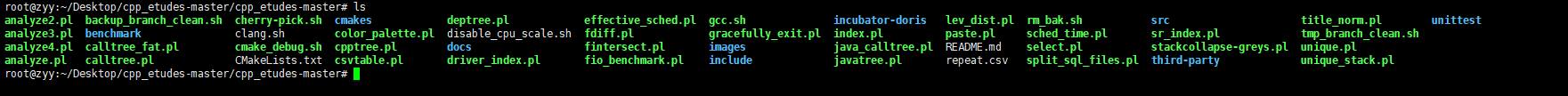
C++阅码神器（基于shell脚本）

cpp\_etudes 是一个与谷歌基准、谷歌测试集成的 cpp 项目，用于验证 cpp 功能和完善数据库的矢量化内置功能。 它可以由 clang 和 gcc 构建， 还包括一些有趣的 perl 脚本。

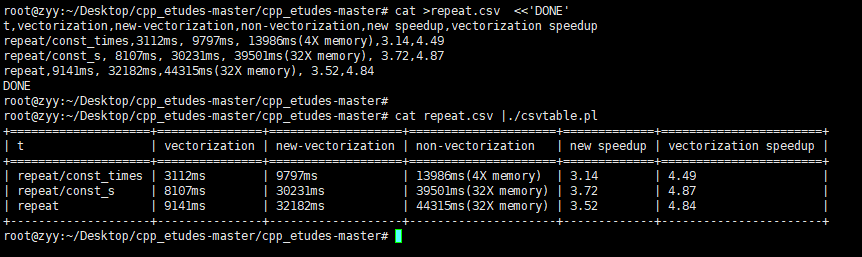
下载：<https://github.com/satanson/cpp_etudes>



解压：



创建一个csv文件，生成一个ascii-border表



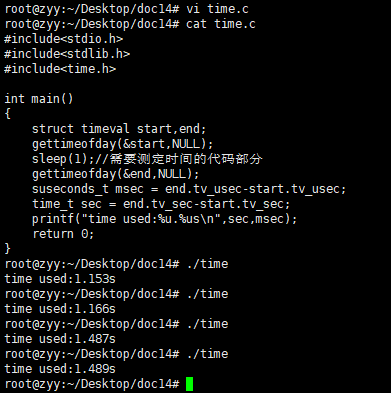
显示颜色鲜艳的终端输出



### (3)第四章 Linux环境下系统函数的使用

使用gettimeofday函数得到程序运行时间

引用头文件time.h，使用gettimeofday()函数



多次运行时间在微秒上会有所差别。

### (4)第五章 文件I/O操作

关于lseek（）函数的用法

#include <errno.h>

#include <fcntl.h>

#include <stdio.h>

#include <stdlib.h>

#include <sys/stat.h>

#include <sys/types.h>

#include <unistd.h>

int main() {

  int fd;

  char str[20] = {"Hello World!"};

  char buf[20] = {0};  //初始化为0,为了方便打印。

  int flag;

  //在当前路径下打开一个new.txt文件，若不存在就创建它；若存在则以可读可写方式打开，并且从长度0截断

  fd = open("./new.txt", O\_CREAT | O\_TRUNC | O\_RDWR, 0777);

  if (-1 == fd) {

    perror("open fail");  //包含在stdio中的函数。

    exit(1);

  }

  flag = write(fd, str, 20);  //给文件写入内容

  if (-1 == flag) {

    perror("write fail");

    exit(1);

  }

  lseek(fd, 0, SEEK\_SET);  //使用lseek将文件偏移量设置为文件开始处

  //如果没有这一步，那么下面的read无法读取到文件内容，因为文件偏移量已经到了字符串结束符处。

  flag = read(fd, buf, 20);  //读取文件内容

  if (-1 == flag) {

    perror("read fail");

    exit(1);

  }

  flag = close(fd);  //关闭文件

  if (-1 == flag) {

    perror("close fail");

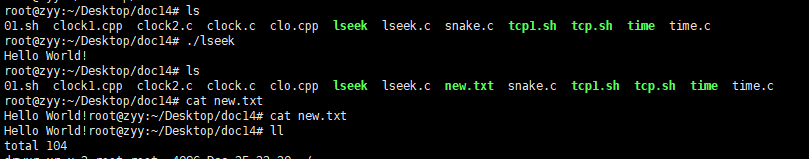
    exit(1);

  }

  printf("%s\n", buf);  //打印缓冲区中的内容。

  return 0;

}



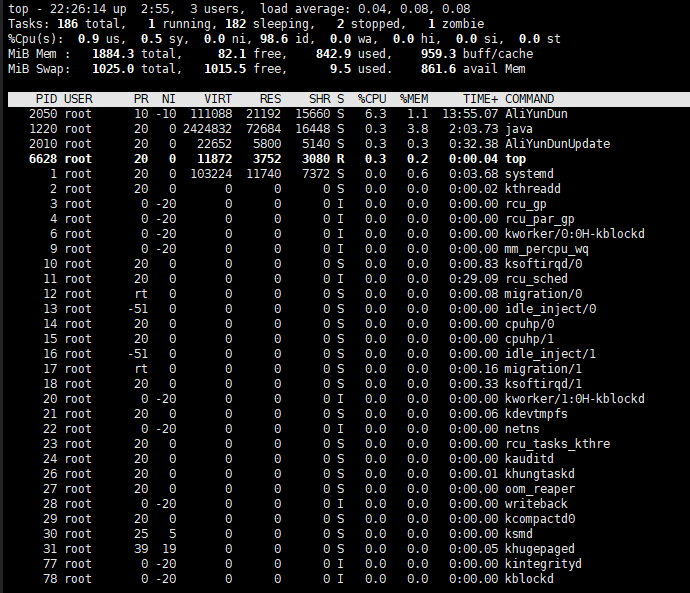
lseek不可以用于管道，FIFO，socket文件。另外lseek的文件偏移量的大小可以大于当前文件的长度，在这种情形下，对该文件的下一次写将加长该文件，并在文件中构成一个空洞。文件空洞并不要求在磁盘上占据空间

### (5)第六章 文件I/O操作

命令TOP

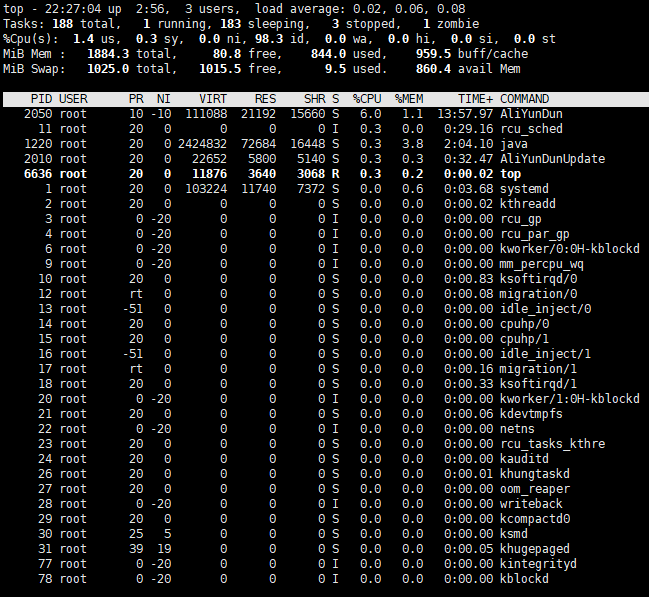
top命令会显示当前系统中占用资源最多的一些进程（默认以CPU占用率排序）如果你想改变排序方式，可以在结果列表中点击O（大写字母O）会显示所有可用于排序的列，这个时候你就可以选择你想排序的列

#top



如果只想显示某个特定用户的进程，可以使用-u选项

#top -u root



### (6)第七章 进程通信

ipcs显示进程通信

选项 说明

-i 显示指定id的信息

-m 显示共享内存段的信息

-q 显示队列信息

-s 显示信号灯信息

-a 显示所有信息，默认选项

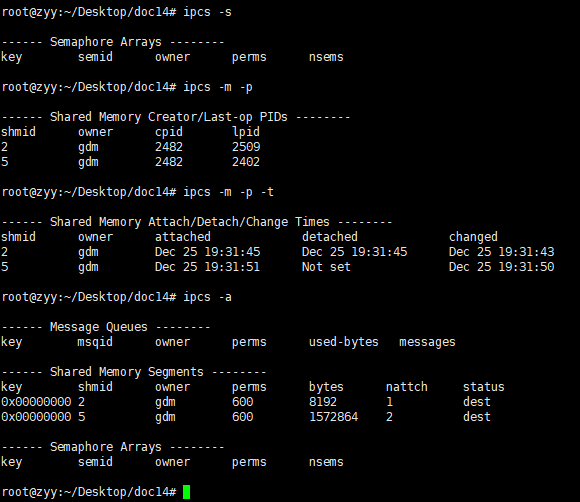
-t 显示便用时间

-p 显示进程pid

-c 显示进控创造者

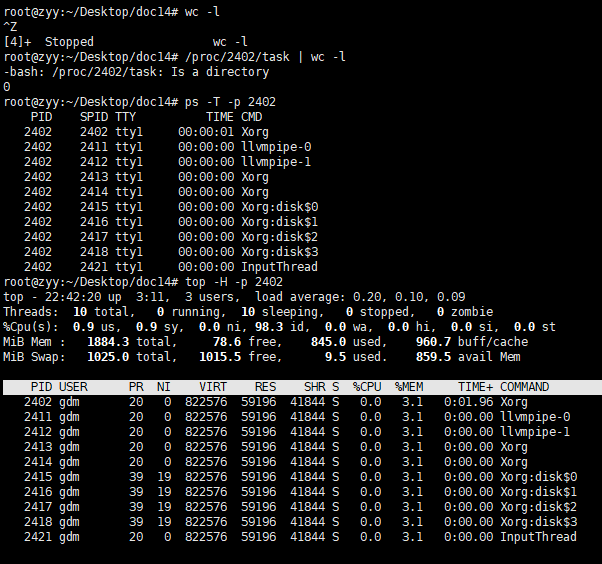
-u 显示总和

-l 显示限制



### (7)第八章 进程通信

查看 Linux 进程的线程数 查看 /proc 信息 ls /proc/<PID>/task | wc -l



### (8)第九章 网络编程

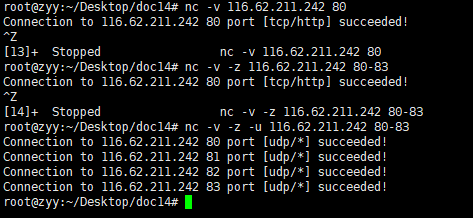
NC命令使用

 linux可以使用nc命令来测试 网络端口是否正常，类似于telnet命令，但也可以用nc命令来监听本地端口，支持TCP、UDP协议，当我们测试NTP服务网络策略是否正常时，可以使用到nc命令测试UDP 123端口。

nc -v ip port #扫瞄某 IP 的某个端口，返回端口信息详细输出。

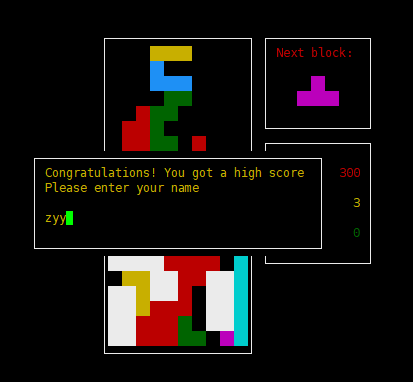
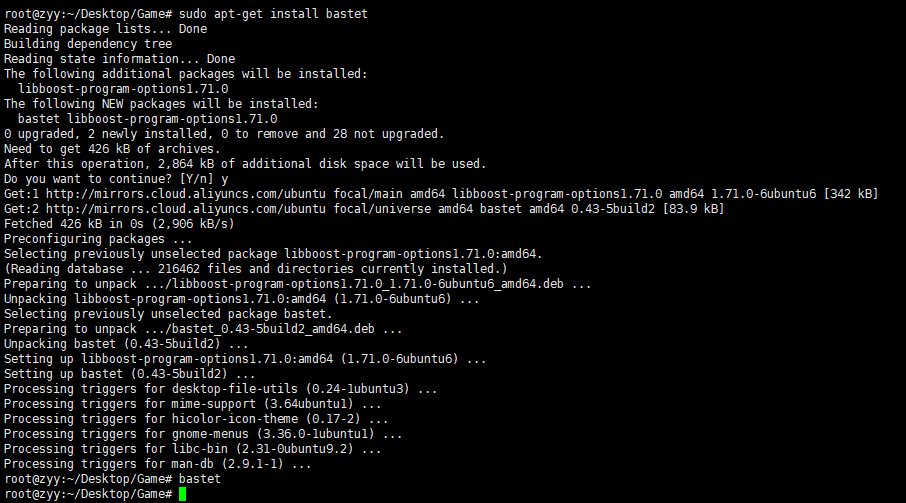
nc -v -z ip port-port #扫描某IP的端口段，返回端口信息详细输出，但扫描速度很慢。

nc -v -z -u ip port-port #扫描某 IP 的某 UDP 端口段，返回端口信息详细输出，但扫描速度很慢。



### (9)第十章 Linux的图形编程

安装俄罗斯方块游戏：<https://www.techug.com/post/10-linux-terminal-games.html>



### (10)第十一章 设备驱动程序设计基础

示当前进程可用的CPU数目：

