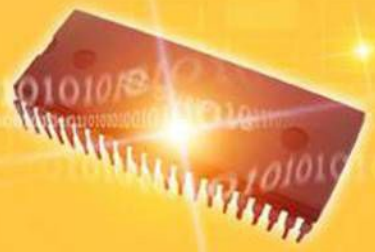
凌阳教育
www.sunplusedu.com
值得信赖的教育品牌

嵌入式系统工程师



Tel: 400-705-9680 , Email: edu@sunplusapp.com , BBS: bbs.sunplusedu.com

Linux开发环境与编程工具

Tel: 400-705-9680 , Email: edu@sunplusapp.com , BBS: bbs.sunplusedu.com

大纲

- linux环境开发概述
- 目录结构及文件
- Linux命令
- 编辑器（vi+gedit）
- 编译器（gcc）
- 调试器（gdb）

大纲

- linux环境开发概述
- 目录结构及文件
- Linux命令
- 编辑器（vi+gedit）
- 编译器（gcc）
- 调试器（gdb）

Linux开发环境概述

- Windows开发特点:
  - 在电脑上装一个vc或其它集成开发环境
  - 编辑程序—>编译程序—>看现象—>有问题—>修改程序->调试程序->查看

- Linux开发特点:
  - Linux下的程序开发大多通过在本地安装虚拟机、物理机或网络连接到服务器完成
  - 出于效率、远程开发、嵌入式开发的考虑：开发方式大多是在命令行下完成，没有很好的集成开发环境供我们使用

大纲

- linux环境开发概述
- 目录结构及文件
- Linux命令
- 编辑器（vi+gedit）
- 编译器（gcc）
- 调试器（gdb）

Linux环境开发概述

- Linux文件及目录结构:
  - 无论何种版本的Linux，桌面、应用是Linux的外衣
  - 文件组织、目录结构才是Linux的内心
  - 其架构上设计的特点:

    大树底下好乘凉：对磁盘分区的管理（树状）

    一切皆文件:　　对数据、硬件设备的管理

Linux目录结构及文件

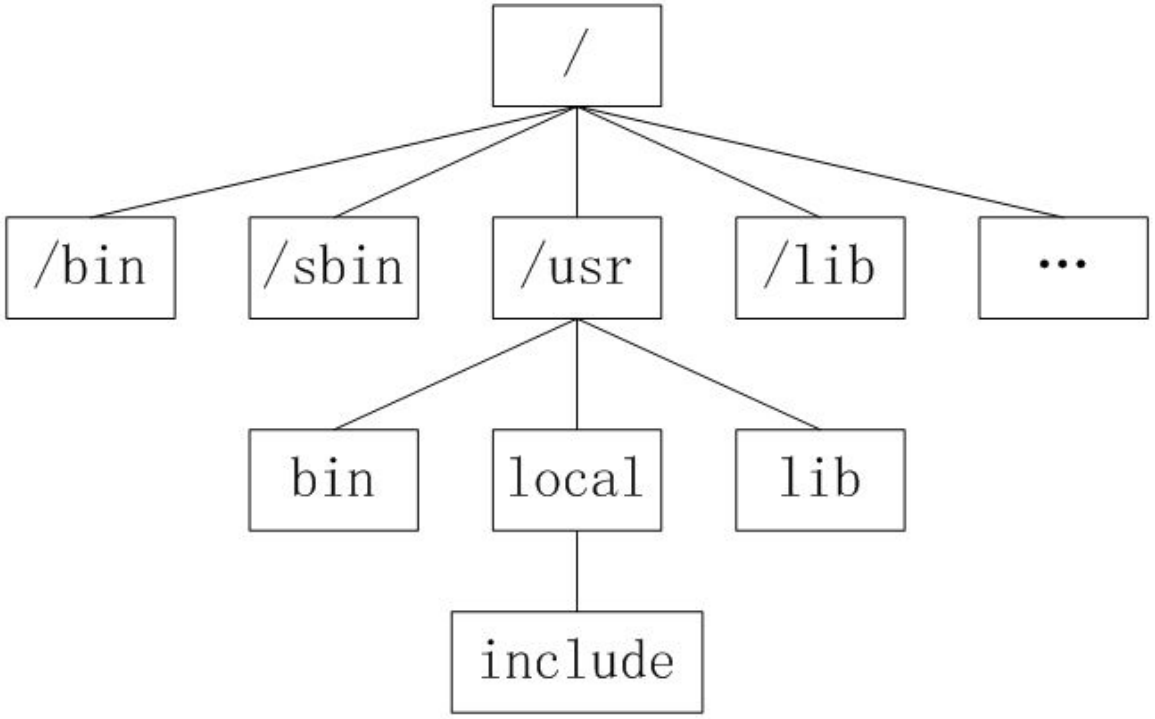
- 大树底下好乘凉——树状结构管理磁盘
  - Linux启动顺序:

    bootloader->内核->根文件系统->应用程序
  - 根文件系统按照一定格式存放在硬盘、FLASH中的某一个分区，Linux内核启动起来后首先会启动它
  - 其他分区、U盘、SD卡、光盘等"挂载"在根文件系统的某一目录下，我们通过此目录访问磁盘
  - 总之:
    - **Linux中看不到类似于windows下的C、D、E**

      我们看到就是一棵大树（rootfs）

Linux目录结构

- Linux目录结构
  - /(根目录)是Linux系统中最顶层的目录，所有的文件夹、文件都是它的子目录。



Linux目录结构及文件

- 一切皆文件:
  - Linux对数据、设备抽象成文件进行管理
  - Linux对:
    - 数据文件(*.mp3、*.bmp)
    - 程序文件(*.c、*.h、*.o)
    - 设备文件(LCD、触摸屏、鼠标)
    - 网络文件(socket)

  等的管理都抽象为文件,使用统一的方式方法管理
  - 总之:
    - **文件在Linux下赋予了崭新的含义和地位**

大纲

- linux环境开发概述
- 目录结构及文件
- Linux命令
- 编辑器（vi+gedit）
- 编译器（gcc）
- 调试器（gdb）

Linux命令概述

- 命令概述
  - Linux刚出世时没有什么图形界面，所有的操作全靠命令完成
  - 近几年来，Linux发展的非常迅速，图形界面越来越友好，但是在真正的开发过程中，Linux命令行的应用还是占有非常重要的席位的
  - 许多Linux功能在命令行界面要比图形化界面下运行的快，有些使用Linux的场合甚至没有图形化的界面
  - 可以说不会命令行，就不算会Linux。

Linux命令使用方法

- 命令使用方法
  - Linux命令格式:

    command [-options] [parameter1] ...
  - 说明:

    command:

    命令名: 相应功能的英文单词或单词的缩写

    [-options]:

    选项: 可用来对命令进行控制, 也可以省略

    parameter1 ...:

    传给命令的参数: 可以是零个一个或多个

    例: ls -alh /home

Tel: 400-705-9680 , Email: edu@sunplusapp.com , BBS: bbs.sunplusedu.com

Linux命令使用方法

- 命令使用方法
  - --help

    一般是Linux命令自带的帮助信息

    如：ls --help
  - man(manual)
    - 有问题找男人
    - man是Linux提供的一个手册，包含了绝大部分的命令、函数使用说明
    - 该手册分成很多章节（section），使用man时可以指定不同的章节来浏览。

  例：man ls ； man 3 printf

  (printf既是命令又是函数，可使用 数字选定章节)

Linux命令使用方法

➢ 命令使用方法

man中各个section意义如下:

1、Standard commands（标准命令）

2、System calls（系统调用，如open,write）

3、Library functions（库函数，如printf,fopen）

Linux命令使用方法

- 命令使用方法:

自动补全:

敲出命令前几个字母的同时按下tab键，系统会帮我们补全命令

历史命令:

当系统执行过一些命令后，可按上下键翻看以前的命令

history将执行过的命令列举出来

重定向: >

本应显示在终端上的内容保存到执行文件中

例: ls > test.txt; ls --help > test.txt

管道: |

一个程序的输出作为另一个程序的输入

ls | less ; ls | more

Linux常用命令

- 常用命令
  - ls tree clear
  - cd pwd
  - mkdir touch
  - cat rm cp mv
  - find grep tar

Linux常用命令

- 常用命令
  - ls：显示目录内容

    -l：列表显示

    -a：显示隐藏文件

    -h：配合-l以人性化的方式显示文件大小
  - tree：以树状形式显示目录结构

    -L n(n为要查看的层数)

    系统默认没有此命令

    安装：sudo apt-get install tree
  - clear:清除终端上的显示

    clear

Linux常用命令

- 常用命令
  - cd：切换目录

    cd /home可进入home（指定）目录

    cd .. 可进入上一层目录

    cd - 可进入上一个进入的目录

    cd ~ 可进入用户的家目录(~：代指家目录)

  - pwd：显示当前工作目录的绝对路径

Linux常用命令

- 常用命令
  - mkdir: 创建目录

    -p 递归地在指定路径建立文件夹
  - touch: 创建文件

    touch test.c
  - cat: 显示文本文件内容

    cat test 显示test文件的内容
  - rm: 删除文件/目录

    rm file1 -rf

    -r: 删除文件夹时必须加此参数

    -f : 强制地删除文件，如果没有不出错

Linux常用命令

- 常用命令
  - cp: 复制文件

    cp /etc/profile /home

    -R: 拷贝文件夹

    -v: 显示拷贝进度

  - mv: 移动文件或更改文件名

    mv file1 file2

    mv file1 /home

    –v: 同cp

Linux常用命令

- 常用命令
  - find: 在指定路径下查找指定文件

    用法: find 路径名-name 文件名

    例子: find /home -name test.txt

    //在根目录的home目录下查找名为test.txt的文件

  - grep: 在指定文件中查找指定内容

    用法: grep 查找信息 文件名 参数-n 显示行号

    例子1: grep aaa test.c -n

    //在当前文件test.c中查找aaa信息，并显示行号

    例子2: grep aaa ./* -R -n

    //在当前文件夹下的所有文件查找包含aaa信息的文件

Linux常用命令

- 常用命令
  - tar: 压缩与解压缩命令

    gzip格式:
    - 压缩用法: tar zcvf 压缩包包名 文件1 文件2 ...

      例子: tar zcvf bk.tar.gz *.c
    - 解压用法: tar xvzf 压缩包包名

      例子: tar zxvf bk.tar.gz
    - 解压到指定目录: -C

      例子: tar zxvf bk.tar.gz -C ./test

      //将bk.tar.gz解压到当前目录下的test目录下

Linux常用命令

- 常用命令
  - tar: 压缩与解压缩命令

    bz2格式:
    - 压缩用法: tar jcvf 压缩包包名 文件1 文件2 ...

      例子: tar jcvf bk.tar.bz2 *.c
    - 解缩用法: tar jxvf 压缩包包名

      例子: tar jxvf bk.tar.bz2
    - 解压到指定目录: -C

      例子: tar jxvf bk.tar.bz2 -C ./

大纲

- linux环境开发概述
- 目录结构及文件
- Linux命令
- 编辑器（vi+gedit）
- 编译器（gcc）
- 调试器（gdb）

gedit编辑器

- **gedit:**

  gedit是一个Linux环境下的文本编辑器

- 类似windows下的写字板程序，在不需要特别复杂的编程环境下，作为基本的文本编辑器比较合适

- 使用:

  在字符界面下，直接使用:

  gedit file1.txt即可

vi编辑器

- vi在Linux界有编辑器之神的美誉

  vi编辑器是Linux系统中最常用的文本编辑器，几乎所有的Linux发行版中都包含vi程序

  vi工作在字符模式下，不需要图形界面，非常适合远程及嵌入式工作,是效率很高的文本编辑器

  尽管在Linux上也有很多图形界面的编辑器可用，但vi的功能是那些图形编辑器所无法比拟的。

vi编辑器

- 1、安装vim

  sudo apt-get install vim

- 2、安装ctags

  sudo apt-get install ctags

- 3、配置vim

  step1：将vim_configure拷入当前用户的目录下

  使用samba或共享文件夹完成

  step2：打开终端，执行以下以下命令：

  cd vim_configure

  ./copy_con.sh

  拷贝成功后，出现copy successful

  （如提示没有权限：请使用:chmod +x copy_con.sh增加执行权限）

vi编辑器

使用vi打开文件

vi filename:

打开或新建文件，并将光标置于第一行行首

vi +n filename:

打开存在文件，并将光标置于第n行行首

vi编辑器

- vim编辑器有3个操作模式:

1、vi插入模式

- vi创建一个不存在文件时，默认进入插入模式
- vi将输入的字符作为正文内容放在正在编辑的文件中

2、vi编辑模式

- vi打开一个已经存在文件时，默认进入编辑模式
- 在此模式下可进入插入模式、控制屏幕光标的移动、进行文本的选择、复制、粘贴、剪切、删除、查找等工作。

3、vi最后一行模式(命令模式)

- 在编辑模式下，按shift+: 进入
- 跟编辑模式类似，完成存盘、另存、查找等任务

vi编辑器

- **由编辑模式切换到插入模式**

  i 从光标当前位置开始插入

  o 在光标位置的下行插入一个空行，再进行插入

- **由编辑模式切换到命令模式**

  shift + :

- **由插入模式、命令模式切换到编辑模式**

  esc

- 插入模式与命令模式不能直接转换

vi编辑器

- 编辑模式下删除和修改文本

1、 u　　撤消前面多次修改。
2、[n]x　　删除光标后n个字符。
3、[n]X　　删除光标前n个字符。
4、[n]dd　　删除从当前行开始的n行。
5、[n]yy　　复制从当前行开始的n行。
6、 p　　把粘贴板上的内容插入到当前行。
7、 .　　执行上一次操作
8、shift +zz(按住shift按两下z键) 保存退出当前文件

vi编辑器

- 编辑模式下移动光标

  [n]G：将光标定位到第n行开始处

  G：将光标定位到文件结束处

  gg:将光标定位到文件开始处

- 编辑模式下的查找

  /字符串：从光标开始处向文件尾查找字符串。

  n：同一方向重复上一次查找命令。

  N：反方向重复上一次查找命令

vi编辑器

- 常用最后一行模式命令
  - 文件存储类:

:w	保存当前文件
:w file	另存当前文件为file
:wq	保存当前文件，退出
:q!	不保存文件并退出

  - 配合搜索命令使用:

:nohls	取消高亮
:set hls	设置高亮

vi编辑器

- 配置文件的作用:
  1、自动添加创建时间、名称等注释信息
  2、自动添加行号
  3、支持鼠标点击定位
  4、支持函数列表功能(F9打开或关闭)
  5、支持多文件打开功能（F5）
  6、可使用ctrl+N 补全函数以及结构体成员
  7、自动整理代码格式:
  使用鼠标选中代码，按下“=”号键可自动规整代码
  8、 支持vim快捷键.pdf中的快捷键。
  插件的特殊用法,参看《vim快捷键.pdf》

大纲

- linux环境开发概述
- 目录结构及文件
- Linux命令
- 编辑器（vi+gedit）
- 编译器（gcc）
- 调试器（gdb）

gcc概述

- 编译器是将易于编写、阅读和维护的高级计算机语言翻译为计算机能解读、运行的低级机器语言的程序。
- GNU项目中的一个子项目GCC（GNU Compiler Collection）是一个编译器套装
- GCC最初用于编译C语言，随着项目的发展GCC已经成为了能够编译C、C++、Java、Ada、fortran、Object C、Object C++、Go语言的编译器大家族。

gcc

- gcc作为真实的编译器和链接器的入口。
  - 它会在需要的时候调用其它组件（预处理器、汇编器、链接器），并且会传一些额外的参数给编译器和连接器。
  - 输入文件的类型和传给gcc的参数决定了gcc调用哪些组件。

gcc

- gcc识别的文件扩展名如下:

.c C语言文件

.i 预处理后的C语言文件

.C、.cc、.cp、.cpp、.c++、.cxx C++语言文件

.ii 预处理后的C++语言文件

.S 汇编文件

.s 预处理后的汇编文件

.o 编译后的目标文件

.a 目标文件的静态链接库（链接时使用）

.so 目标文件的动态链接库（链接、运行时使用）

gcc

- gcc、g++编译选项

-o file	指定生成的输出文件名为file
-E	只进行预处理
-S	只进行预处理和编译
-c	只进行预处理、编译和汇编
-Wall	生成所有级别的警告信息
-w	关闭所有警告，建议不使用此选项

gcc

- gcc常用编译应用实例:
- 例1: 分步进行

```
gcc -E hello.c -o hello.i     1、预处理
gcc -S hello.i -o hello.s     2、编译
gcc -c hello.s -o hello.o     3、汇编
gcc    hello.o -o hello_elf   4、链接
```

- 例2: 一步到位

```
gcc    hello.c -o hello_elf
```

大纲

- linux环境开发概述
- 目录结构及文件
- Linux命令
- 编辑器（vi+gedit）
- 编译器（gcc）
- 调试器（gdb）

gdb简介

- GNU工具集中的调试器是gdb，该程序是一个交互式工具，工作在字符模式。
- 除gdb外，linux下比较有名的调试器还有xxgdb, ddd, kgdb, ups。

gdb的使用

- gdb是功能强大的调试器，可完成如下调试任务:
  1、设置断点
  2、监视程序变量的值
  3、程序的单步执行
  4、显示/修改变量的值
  5、显示/修改寄存器
  6、查看程序的堆栈情况
  7、远程调试

gdb的使用

GDB常用命令：

```
1、l（list）--------------列出程序清单
2、r（run）---------------运行程序
3、b [行号\函数名]--------设置断点,b（breakpoint）
4、info b-----------------查看断点信息
5、clear [行号]-----------清除断点
6、c（continue）----------继续运行程序
7、s（step）--------------单步跳入运行 step into
8、n（next）--------------单步跳过运行 step over
9、finish----------------跳出函数   step out
10、print   变量\表达式---显示变量或表达式的值
11、display 变量\表达式 --每次运行停止都打印显示变量表达式的值
12、undisplay 标号--------清除标号对应的变量值的显示
```

gdb的使用

- 下面以一个小程序来说明gdb的使用流程

```
1  #include <stdio.h>
2
3  int sum(int m)
4  {
5      int i = 0, n = 0;
6
7      for ( i = 1; i <= m; i += 1 )
8          n += i;
9
10     printf("The sum of 1-m is %d\n", n);
11
12     return 0;
13 }
14
15 int main(int argc, char *argv[])
16 {
17     int i = 0, n = 0;
18     sum(50);
19
20     for ( i = 1; i <= 50; i += 1 )
21         n += i;
22
23     printf("The sum of 1-50 is %d\n", n);
24     return 0;
25 }
26
27
```

gdb的使用

```
[gdb_test]gcc -g  test.c  -o test
[gdb_test]gdb test
GNU gdb (Ubuntu/Linaro 7.4-2012.04-0ubuntu2.1) 7.4-2012.04
Copyright (C) 2012 Free Software Foundation, Inc.
License GPLv3+: GNU GPL version 3 or later <http://gnu.org/licenses/gpl.html>
This is free software: you are free to change and redistribute it.
There is NO WARRANTY, to the extent permitted by law.  Type "show copying"
and "show warranty" for details.
This GDB was configured as "i686-linux-gnu".
For bug reporting instructions, please see:
<http://bugs.launchpad.net/gdb-linaro/>...
Reading symbols from /home/delong/tmp/gdb_test/test...done.
(gdb) l
8                       n += i;
9
10              printf("The sum of 1-m is %d\n", n);
11
12              return 0;
13      }
14
15      int main(int argc, char *argv[])
16      {
17              int i = 0, n = 0;
(gdb) list
18              sum(50);
19
20              for ( i = 1; i <= 50; i += 1 )
21                      n += i;
22
23              printf("The sum of 1-50 is %d\n", n);
24              return 0;
25      }
26
```

gdb的使用

```
(gdb) break 7
Breakpoint 1 at 0x80483f8: file test.c, line 7.
(gdb) info break
Num     Type           Disp Enb Address    What
1       breakpoint     keep y   0x080483f8 in sum at test.c:7
(gdb) run
Starting program: /home/delong/tmp/gdb_test/test

Breakpoint 1, sum (m=50) at test.c:7
7               for ( i = 1; i <= m; i += 1 )
(gdb) print i
$1 = 0
(gdb) p i
$2 = 0
(gdb) step
8                       n += i;
(gdb) continue
Continuing.
The sum of 1-m is 1275
The sum of 1-50 is 1275
[Inferior 1 (process 6752) exited normally]
(gdb)
```

作业

- 完成：linux常用命令_练习.txt
- 完成：vi、gcc、gdb练习.txt

凌阳教育
www.sunplusedu.com

值得信赖的教育品牌

Tel: 400-705-9680 , Email: edu@sunplusapp.com , BBS: bbs.sunplusedu.com