ls -a 所有文件

ls -l 详细属性

more一页一页显示，更多参数

tail -f 查阅正在改变的文件

uptime 时间 系统运行时长 负载

uname 操作系统信息

locate file 在数据库内找，比find快

ctrl+c 中断，无后台

ctrl+z 暂停，后台挂起

tar flags -T 范本文件，让tar解开的文件

-v或--verbose 显示指令执行过程

-c或--create 建立新的备份文件

-t或--list 列出备份文件的内容

-w或--interactive 遭遇问题时先询问用户

-x或--extract或--get 从备份文件中还原文件

-f<备份文件>或--file=<备份文件> 指定备份文件

-z或--gzip或--ungzip 通过gzip指令处理备份文件

-k或--keep-old-files 解开备份文件时，不覆盖已有的文件

Whois 在数据库找信息

Dig 域名查询

Dig [@server] name type ,server 是域名服务器的名称或 IP 地址, type 指示所需的查询类型 ANY、A、MX、SIG 等

Dig -x 逆向域名查询

Extra commands

PATH - colon separated list of pathnames to search for commands

GCC 的编译过程可以划分为四个阶段：预处理（Pre-Processing）、编译（Compiling）、汇编（Assembling）以及链接（Linking）

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| |  |  |  |  | | --- | --- | --- | --- | | .c | C 源文件 | .s/.S | 汇编语言源文件 | | .C/.cc/.cxx/.cpp | C++ 源文件 | .o/.obj | 目标文件 | | .h | C/C++ 头文件 | .a/.lib | 静态库 | | .i/.ii | 经过预处理的 C/C++ 文件 | .so/.dll | 动态库 | |

-pipe ：使用管道而不是中间文件

-E ：仅执行预处理（不要编译、汇编或链接）。

-S ：只编译（不汇编或链接）。

-c ：编译和汇编，但不链接。

-o <file> ：指定输出文件。

-g 编译时额外：1. 创建调试符号表，符号表包含了程序中使用的变量名称的列表。

2. 关闭所有的优化机制，以便程序执行过程中严格按照原来的C代码进行

如果没有可执行文件中的调试信息，你就无法在gdb中看到代码。在编译的时候，生成调试信息，该程序可以被调试器调试

/

编译 hello.c，默认输出 a.out

gcc hello.c

/

编译 hello.c 并指定输出文件为 hello

gcc hello.c -o hello

/

只执行预处理，输出 hello.i 源文件

gcc -E hello.c -o hello.i

/

只执行预处理和编译，输出 hello.s 汇编文件

gcc -S hello.c

/

也可以由 hello.i 文件生成 hello.s 汇编文件

gcc -S hello.i -o hello.s

/

只执行预处理、编译和汇编，输出 hello.o 目标文件

gcc -c hello.c

/

也可以由 hello.i 或 hello.s 生成目标文件 hello.o

gcc -c hello.i -o hello.o

gcc -c hello.s -o hello.o

/

由 hello.o 目标文件链接成可执行文件 hello

gcc hello.o -o hello

/

查看所有文件描述

$ file \*

/

查看 hello 文件描述

$ file hello

/

把f1上传到svn上步骤，Svn add f1; svn commit f1; svn revert f1

Revert是回溯到原版本

显示的命令放入到xxx.txt 文件中 ls > xxx.txt

显示的命令追加到xxx.txt中 ls>>xxx.txt

清屏 clear

cd . 当前路径

cd .. 上级目录

cd ../.. 上级目录的上级目录

cd - 回到上次所在路径

cd~ 回根目录

grep "ntfs" xxx.txt

grep -n "ntfs" xxx.txt 在哪一行

grep -v "ntfs" xxx.txt 不包括ntfs的行

grep "^ntfs" xxx.txt 以ntfs开头的

grep "ntfs$" xxx.txt 以ntfs结尾的

mv 111.txt laowang/ 移动到laowang/文件夹中

find / -name "name"

sudo find / -name "name" 获取权限

sudo 是允许系统管理员让普通用户执行一些或者全部的root命令的一个工具

关机 shutdown -h

重启 reboot

同时打开两个终端 Ctrl+Shift +t

su shuaige 切换到帅哥账号

sudo passwd shuaige 设置帅哥的密码

GCC 的编译过程可以划分为四个阶段

写后缀：

c源文件

c++源文件

头文件

预处理后

汇编

目标

静态库

动态库

使用管道而不是中间文件

仅执行预处理（不要编译、汇编或链接）

只编译（不汇编或链接）

编译和汇编，但不链接

指定输出文件

编译时额外：1. 创建调试符号表2. 关闭所有的优化机制

编译 hello.c，默认输出 a.out

编译 hello.c 并指定输出文件为 hello

只执行预处理，输出 hello.i 源文件

只执行预处理和编译，输出 hello.s 汇编文件

也可以由 hello.i 文件生成 hello.s 汇编文件

只执行预处理、编译和汇编，输出 hello.o 目标文件

也可以由 hello.i 或 hello.s 生成目标文件 hello.o

由 hello.o 目标文件链接成可执行文件 hello

查看所有文件描述

查看 hello 文件描述

把f1上传到svn上步骤

显示的命令放入到xxx.txt 文件中

显示的命令追加到xxx.txt中

清屏

当前路径

上级目录

上级目录的上级目录

回到上次所在路径

回根目录

从xxx.txt获取"ntfs"

在哪一行

不包括ntfs的行

以ntfs开头的

以ntfs结尾的

把1.txt移动到laowang/文件夹中

找文件

找文件 获取权限

关机

重启

同时打开两个终端

切换到aa账号

设置aa的密码

c