X删除一个字符 #x删除几个字符 dw删除一个单词 #dw删除几个单词

Dd删除一行 #dd删除多行 d$删除光标到行尾内容 J清除光标所处的行与上一行之间的空格 使用u复原此前的情况

/search正向查找 ?search反向查找 :s /search/replace/g将当前光标所处的行中的search单词替换成replace :%s /search/replace把文档中所有search替换成replace

:#,# s/search/replace/g表示从多少行道多少行，把search替换成replace

使用gcc hello.c -o hello编译hello.c 使用./hello执行

使用gcc -E hello.c -o hello.i进行预处理，生成hello.i文件

使用gcc -S hello.i -o hello.s进行编译，生成汇编文件hello.s

使用gcc -c hello.s -o hello.o对hello.s进行汇编生成目标文件hello.o

使用gcc hello.o -o hello将hello.o链接库生成可执行文件

Gcc hello.c -o -static hello静态链接库

Gcc -g hello.c -o hello生成包含调试信息的文件，使用gdb调试器进行调试

使用-D在编译时进行宏定义gcc -DLINUX hello.c -o hello

#makefile test for hello.c

.PHONY: all clean

CC = gcc

CFLAGS = -wall -g

all: hello

hello: hello.o

$(CC) $(CFLAGS) hello.o -o hello

hello.o: hello.c

$(CC) $(CFLAGS) -c hello.c -o hello.o

clean:

@echo”Begin remove…”

rm -f hello \*.o

编译多个文件

main: main.o mytool1.o mytool2.o

gcc -o main mian.o mytool1.o mytool2.o

mian.o: main.c mytool1.h mytool2.h

gcc -c -g main.c

mytool1.o:mytool1.c mytool1.h

gcc -c -g mytool1.c

mytool2.o:mytool2.c mytool2.h

gcc -c -g mytool2.c

su利用root用户登录到系统

用pwd命令查看当前所在的目录

用ls命令列出此目录下的所有文件和目录

用-a选项列出此目录下包括隐藏文件在内的所有文件和目录

使用touch命令在当前目录创建一个新的文件

使用cp /etc/profile ./ 复制系统文件到当前目录下

创建一个新用户user01 useadd -d /home/user01 -m user01

给用户user01设置密码passwd user01

锁定用户user01 passwd –l user01

解除对用户user01的锁定passwd -u user01

更改用户user01的账户名为user02 usermod -l user02 user01

删除用户userdel user01

创建一个新组stuff:goupadd stuff

创建一个新账户user02，并把他的主组和附属组都设为stuff useradd -g stuff -G stuff user02

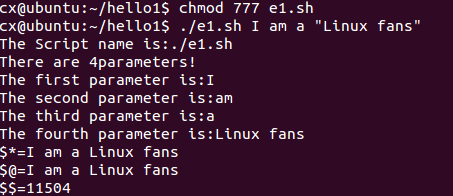
给stuff设置组密码 gpasswd stuff

在组stuff中删除用户user02 gpasswd -d use02 stuff

删除组stuff groupdel stuff 如不能删除更改用户user02的主组usermod -g user01 user02

tar -cvf abc.tar abc将abc打包成abc.tar

tar -xvf abc.tar将abc.tar解包

tar rvf abc.tar d将d追加到abc.tar中

tar -zcvf ok.tar.gz ok将ok目录进行打包并压缩

tar -zxvf ok.tar.gz将ok解压缩出来

gzip file将file压缩成.gz文件

gzip -d file.gz将file.gz文件解压缩

使用free命令显示内存的使用情况

使用df命令显示系统的硬盘分区及使用状况

设置网卡eth1的ip地址为10.22.1.103、掩码为255.255.255.0 ifconfig eht1 10.22.1.103 netmask 255.255.255.0

禁用网卡ifdown eth1

显示网络连接、路由表信息netstat

向本机发送icmp echo request包ping localhost

使用tab键使用命令补齐功能

命令别名功能alias ls = ‘ls -l’

输出重定向ls >> list 输入重定向mail root<list

利用管道和grep命令，在上面建立的文件list中茶包字符串list:#cat list | grep list

查看系统所有的环境变量: echo $PATH

设置环境变量PATH的值，把当前目录加入到命令搜索路径中去PATH”$PATH:.”

#!/bin/sh

echo "The Script name is:$0"

echo "There are $# parameters!"

echo "The first parameter is:$1"

echo "The second parameter is:$2"

echo "The third parameter is:$3"

echo "The fourth parameter is:$4"

echo '$\*='$\*

echo '$@='$@

echo '$$='$$

验证字符型设备文件：

#！/bin/bash

read –p “please enter your filename:” FILENAME

if [-c $FILENAME ];then

cp $FILENAME /dve

else echo “The $FILENAME is not c file”

fi

写个从用户接受一个文件名和一个目录名的shell脚本

#！/bin/bash

read –p “please enter your filename:” FILENAME

if [-e $FILENAME ];then

cp $FILENAME /dve

else echo “The $FILENAME is not c exist”

fi

打印边长为 n 的由“\*”号组成的等边三角形：

#!/bin/bash

if [-z $1];then

echo “please provide a number”

exit

fi

for((i=1;j<$1;i++));do

for((j=0;j<$1-i;j++));do

echo –n “”

done

for((j=0;j<I;j++));do

echo –n “\*”

done

echo””

done

解释程序

fd=open(“fileopen.text”,o\_weonly|o\_creat,0646)

**创建一个fileopen.txt文件，并只写打开，创建错误返回create file wrong！**

**然后将tem加一写入fileopen.txt**

|  |  |
| --- | --- |
| 命令 | 功能 |
| cat | 显示文件的内容(经常和more搭配使用) |
| chgrp | 改变文件或目录所属的用户组 |
| chown | 改变文件的属权 |
| clear | 清除终端屏幕 |
| cp | 将文件或目录复制到其他目录中 |
| diff | 找出两个文件的不同之处 |
| find | 在目录中搜索文件，并执行指定的操作 |
| head | 只查看文件的头几行内容 |
| ln | 在文件之间创建链接 |
| ls | 显示当前目录中的文件和子目录列表 |
| mkdir | 建立子目录 |
| rmdir | 删除“空”的子目录或无用的目录文件 |
| mv | 将文件以及目录移到其他位置，或更改文件以及目录的名称 |
| pwd | 显示用户当前所在的目录 |
| rm | 删除系统中过时或无用的文件 |
| man | 详细了解某个命令 |

设计程序，获取passwd文件大小

#include<sys/stat.h>

#include<unistd.h>

int main(){

struct stat buf;

stat(“/et/passwd”,&buf)

printf(“/etc/passwd file size is:%d bytes size\n”,buf.st\_size)

return 0;}

**把系统中“/etc”目录下的passwd文件权限，设置成文件所有者可读可写，所有其他用户为只读权限**

#include<sys/stat.h>

#include<unistd.h>

int main(){

chmod(“/etc/passwd”,S\_IRUSR|S\_IWUSR|S\_IRGRP|S\_IROTH);

return 0;

}

显示本用户进程#ps 显示所有用户的进程#ps –au 在后台运行cat命令#cat& 查看进程cat #ps aux |grep cat 杀死进程cat

#kill -9 cat执行命令cat

按【ctrl+z】挂起进程

输入jobs命令，查看作业

输入bg，把cat切换到后台执行

输入fg，把cat切换到前台执行

按【ctrl+c】结束进程

获取当前进程及父进程的进程ID

#include <stdio.h>#include <sys/types.h>

#include <unistd.h>void main(){ pid\_t my\_pid, parent\_pid; my\_pid = getpid(); parent\_pid = getppid(); printf("Process id:%ld\n",(long)my\_pid); printf("Parent id:%ld\n",(long)parent\_pid);}

**实现父进程输出打印“This is the parent”3次，子进程输出打印“This is the child”5次。**

#include <sys/types.h>

#include <unistd.h>

#include <stdio.h>

#include <stdlib.h>

int main()

{ pid\_t pid;

char \*message;

int n = 2;

printf("fork program starting\n");

pid = fork();

switch(pid)

{

case -1:

perror("fork failed");

exit(1);

case 0:

message = "This is the child";

n = 5;

break;

default:

message = "This is the parent";

n++;

break;

}

for(; n > 0; n--) {

puts(message);

sleep(1);

}

exit(0);

}

该程序的名字为hello，使用execl()调用该hello程序

#include <unistd.h>#include <stdio.h>#include <stdlib.h> int main(int argc,char \*argv[]){ if(argc<2) { perror("you haven,t input the filename,please try again!\n");

exit(EXIT\_FAILURE);

} if(execl("./hello","hello",0,NULL)<0) perror("execl error!"); return 0;}