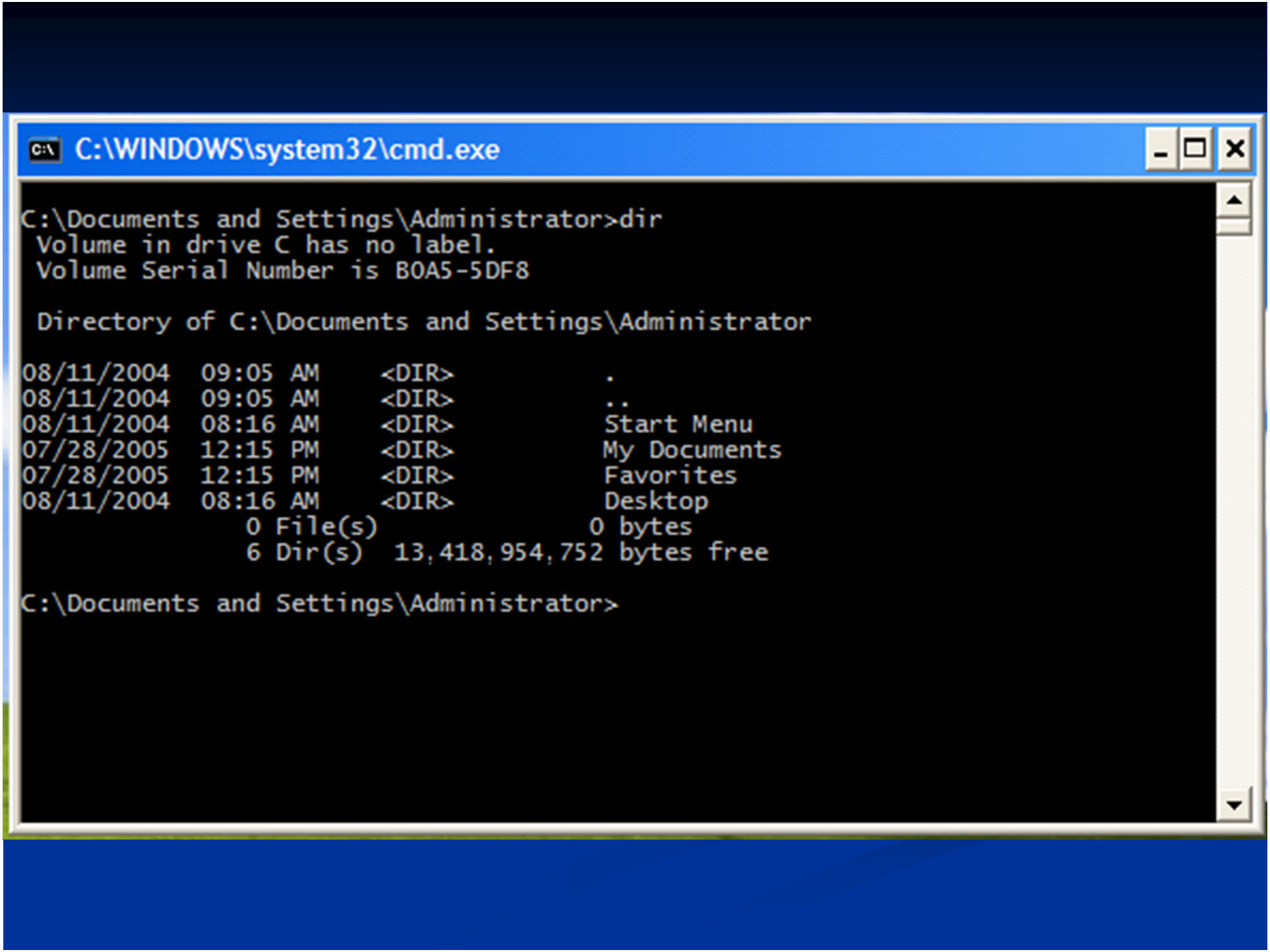
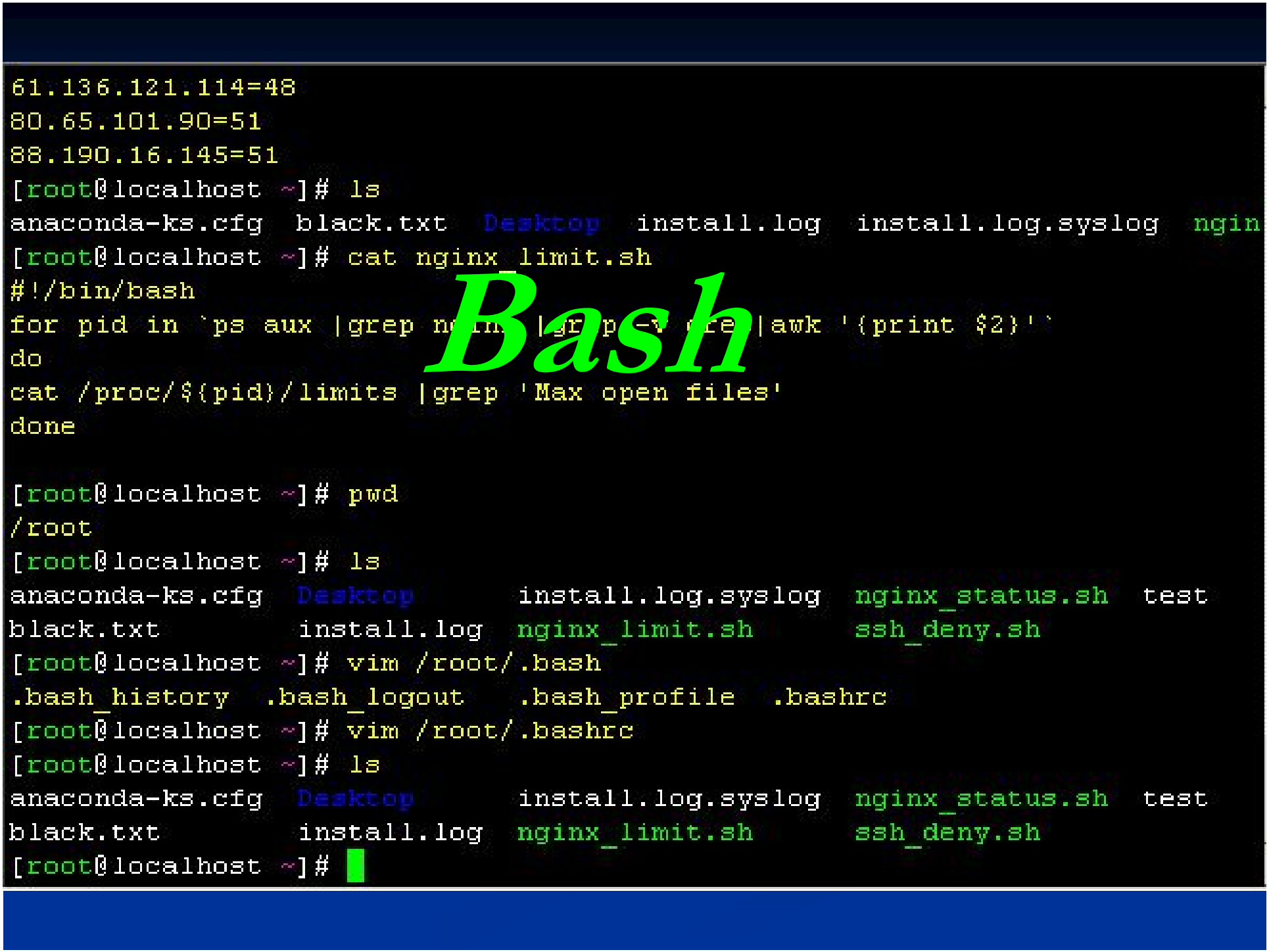
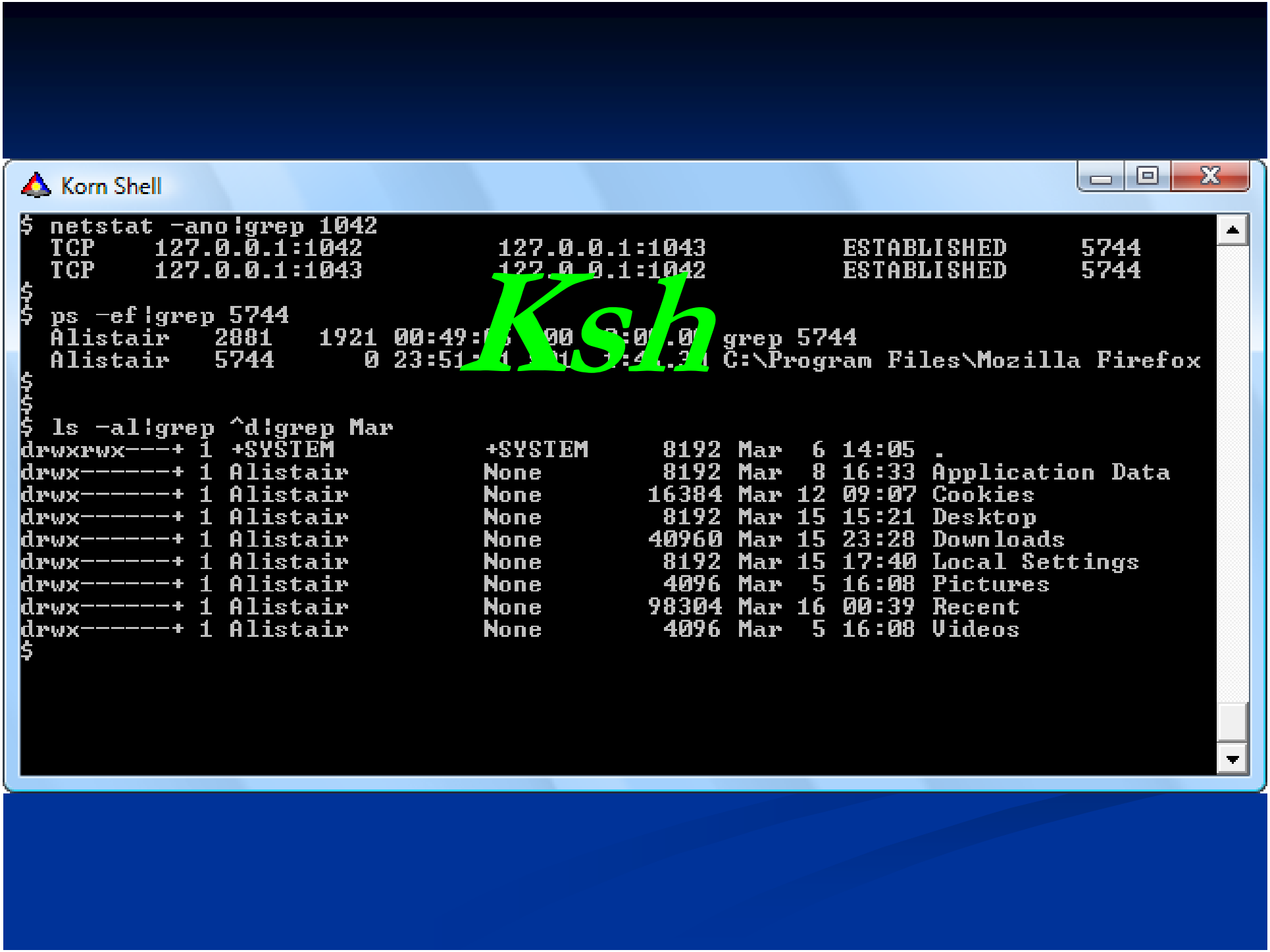
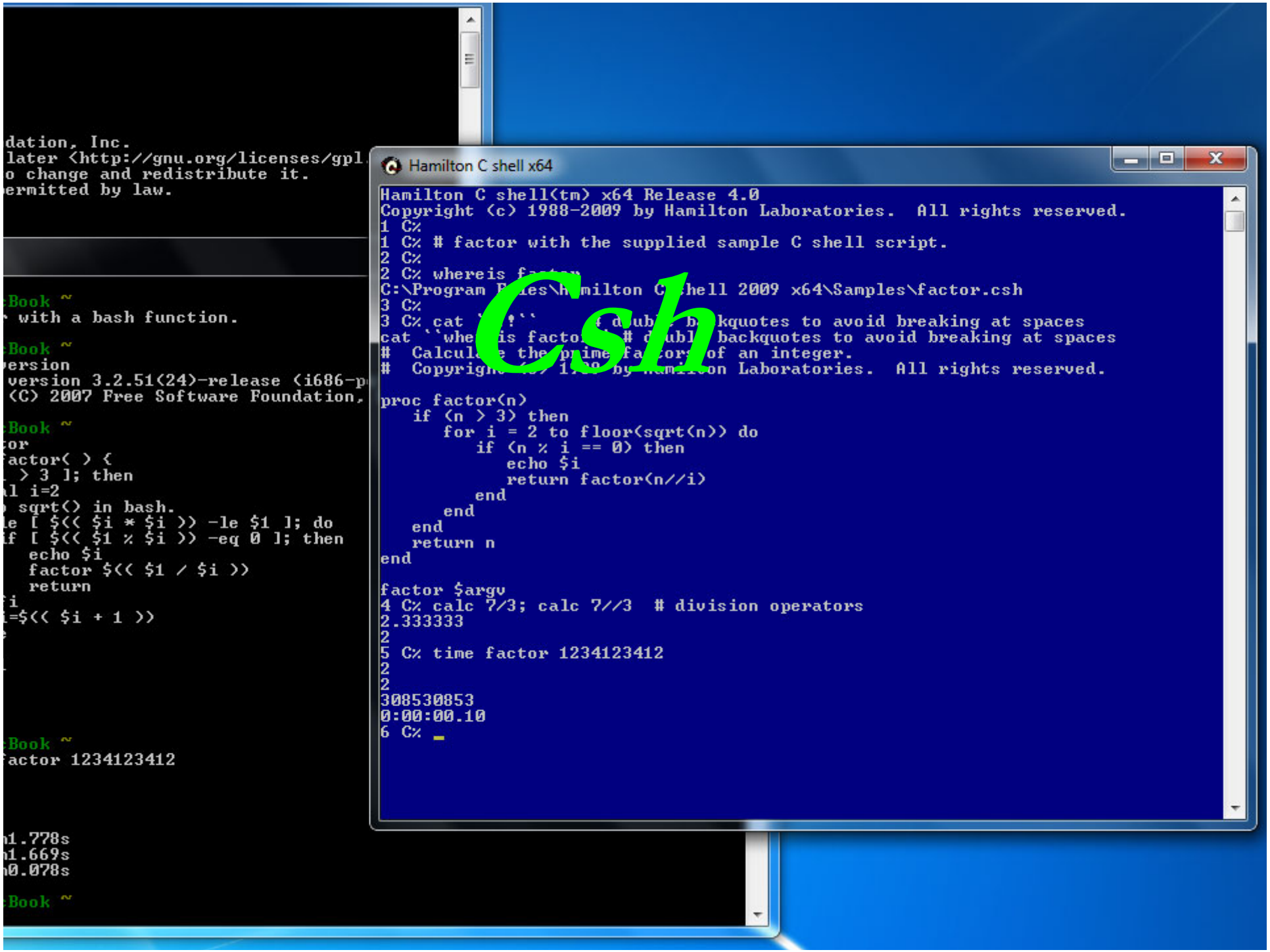
Shell


C:\WINDOWS\system32\cmd.exe
C:\Documents and Settings\Administrator>dir
Volume in drive C has no label.
Volume Serial Number is B0A5-5DF8
Directory of C:\Documents and Settings\Administrator
08/11/2004 09:05 AM <DIR> .
08/11/2004 09:05 AM <DIR> ..
08/11/2004 08:16 AM <DIR> Start Menu
07/28/2005 12:15 PM <DIR> My Documents
07/28/2005 12:15 PM <DIR> Favorites
08/11/2004 08:16 AM <DIR> Desktop
0 File(s) 0 bytes
6 Dir(s) 13,418,954,752 bytes free
C:\Documents and Settings\Administrator>

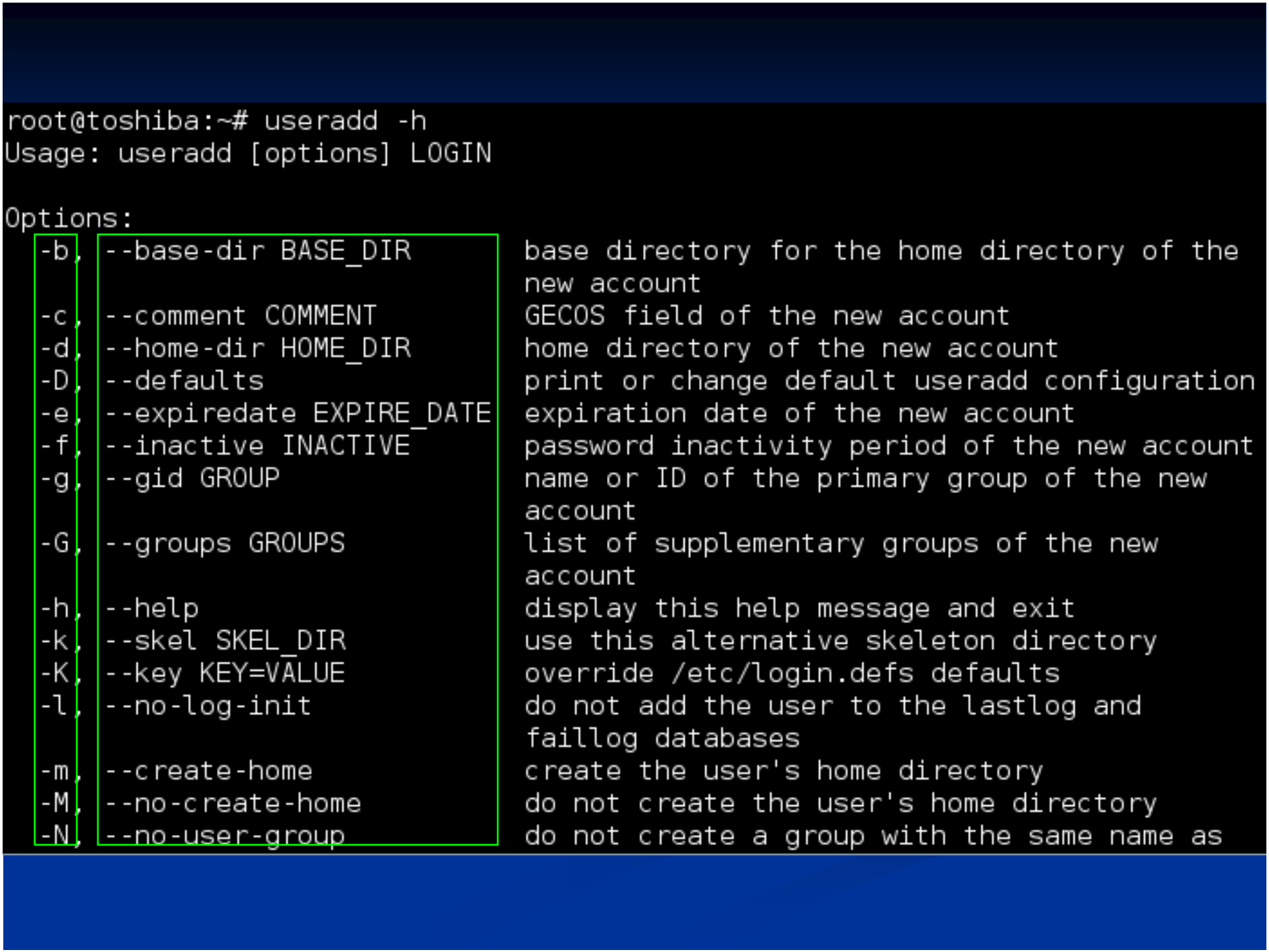

61.136.121.114=48
80.65.101.90=51
88.190.16.145=51
[root@localhost ~]# ls
anaconda-ks.cfg black.txt Desktop install.log install.log.syslog ngin
[root@localhost ~]# cat nginx_limit.sh
#!/bin/bash
for pid in `ps aux |grep n |grep -v grep|awk '{print $2}'`
do
cat /proc/${pid}/limits |grep 'Max open files'
done
[root@localhost ~]# pwd
/root
[root@localhost ~]# ls
anaconda-ks.cfg Desktop install.log.syslog nginx_status.sh test
black.txt install.log nginx_limit.sh ssh_deny.sh
[root@localhost ~]# vim /root/.bash
.bash_history .bash_logout .bash_profile .bashrc
[root@localhost ~]# vim /root/.bashrc
[root@localhost ~]# ls
anaconda-ks.cfg Desktop install.log.syslog nginx_status.sh test
black.txt install.log nginx_limit.sh ssh_deny.sh
[root@localhost ~]#
Bash


Korn Shell
$ netstat -ano|grep 1042
TCP 127.0.0.1:1042 127.0.0.1:1043 ESTABLISHED 5744
TCP 127.0.0.1:1043 127.0.0.1:1042 ESTABLISHED 5744
$
$ ps -ef|grep 5744
Alistair 2881 1921 00:49:
Alistair 5744 0 23:51:
grep 5744
C:\Program Files\Mozilla Firefox
$
$ ls -al|grep ^d|grep Mar
drwxrwx---+ 1 +SYSTEM +SYSTEM 8192 Mar 6 14:05 .
drwx------+ 1 Alistair None 8192 Mar 8 16:33 Application Data
drwx------+ 1 Alistair None 16384 Mar 12 09:07 Cookies
drwx------+ 1 Alistair None 8192 Mar 15 15:21 Desktop
drwx------+ 1 Alistair None 40960 Mar 15 23:28 Downloads
drwx------+ 1 Alistair None 8192 Mar 15 17:40 Local Settings
drwx------+ 1 Alistair None 4096 Mar 5 16:08 Pictures
drwx------+ 1 Alistair None 98304 Mar 16 00:39 Recent
drwx------+ 1 Alistair None 4096 Mar 5 16:08 Videos
$
Ksh


dation, Inc.
later <http://gnu.org/licenses/gpl.
o change and redistribute it.
ermitted by law.
Book ~
with a bash function.
Book ~
version
version 3.2.51(24)-release (i686-p
(C) 2007 Free Software Foundation,
Book ~
or
factor( ) {
> 3]; then
l i=2
sqrt() in bash.
e [$(($i * $i)) -le $1]; do
f [$(($1 % $i)) -eq 0]; then
echo $i
factor $(($1 / $i))
return
i
=$(($i + 1))
Book ~
actor 1234123412
1.778s
1.669s
0.078s
Book ~
Hamilton C shell x64
Hamilton C shell(tm) x64 Release 4.0
Copyright (c) 1988-2009 by Hamilton Laboratories. All rights reserved.
1 C%
1 C% # factor with the supplied sample C shell script.
2 C%
2 C% whereis factor
C:\Program Files\Hamilton C Shell 2009 x64\Samples\factor.csh
3 C%
3 C% cat `!!`
cat `whereis factor` # double backquotes to avoid breaking at spaces
Calculate the prime factors of an integer.
Copyright (c) 1988-2009 by Hamilton Laboratories. All rights reserved.
proc factor(n)
if (n > 3) then
for i = 2 to floor(sqrt(n)) do
if (n % i == 0) then
echo $i
return factor(n//i)
end
end
end
return n
end
factor $argv
4 C% calc 7/3; calc 7//3 # division operators
2.333333
2
5 C% time factor 1234123412
2
2
308530853
0:00:00.10
6 C%
Csh


pcnet32.c:v1.27a 10.02.2002 tsbogend@alpha.franken.de
pcnet32: PCnet/PCI II 79C970A at 0x10e0, 00 0c 29 43 b8 26 assigned IRQ 18.
eth0: registered as PCnet/PCI II 79C970A
pcnet32: PCnet/PCI II 79C970A at 0x1400, 00 0c 29 43 b8 30 assigned IRQ 19.
eth1: registered as PCnet/PCI II 79C970A
pcnet32: 2 cards_found.
Starting interface eth0 [OK]
Starting interface eth1 [OK]
NET4: Ethernet Bridge 008 for NET4.0
Bridge firewalling registered
eth0: Promiscuous mode enabled.
device eth0 entered promiscuous mode
eth1: Promiscuous mode enabled.
device eth1 entered promiscuous mode
Starting interface br0br0: port 2(eth1) entering learning state
br0: port 1(eth0) entering learning state
[OK]
Enable Routing [OK]
Disabling ECN [OK]
Starting WLAN Devices: [OK]
Starting Firewall [OK]
Starting system log daemon [OK]
Starting kernel log daemon [OK]
Starting hotplug modules
Setting GRSecurity parameters: [OK]
键入你的用户名
Welcome to Devil-Linux v1.1.1-2003-12-31-i586-SMP - Kernel 2.4.23-grsec (tty1).
Devil login:


ubuntu
Username:
softpedia
SOFTPEDIA
www.softpedia.com
Options
softpedia-desktop // Wed Apr 25, 8:23 AM


root@toshiba:~# useradd -h
Usage: useradd [options] LOGIN
Options:
-b, --base-dir BASE_DIR base directory for the home directory of the new account
-c, --comment COMMENT GECOS field of the new account
-d, --home-dir HOME_DIR home directory of the new account
-D, --defaults print or change default useradd configuration
-e, --expiredate EXPIRE_DATE expiration date of the new account
-f, --inactive INACTIVE password inactivity period of the new account
-g, --gid GROUP name or ID of the primary group of the new account
-G, --groups GROUPS list of supplementary groups of the new account
-h, --help display this help message and exit
-k, --skel SKEL_DIR use this alternative skeleton directory
-K, --key KEY=VALUE override /etc/login.defs defaults
-l, --no-log-init do not add the user to the lastlog and faillog databases
-m, --create-home create the user's home directory
-M, --no-create-home do not create the user's home directory
-N, --no-user-group do not create a group with the same name as

```
root@toshiba:~# useradd -d /home/kaseihi -s /bin/bash kaseihi
```

/etc/passwd

```
statd:x:102:65534::/var/lib/nfs:/bin/false
avahi-autoipd:x:103:106:Avahi autoip daemon,,,:/var/lib/avahi-autoipd:/bin/false
messagebus:x:104:107::/var/run/dbus:/bin/false
sshd:x:105:65534::/var/run/sshd:/usr/sbin/nologin
kaseihi:x:1000:1000:Jia Jingping,,,:/home/kaseihi:/bin/bash
usbmux:x:106:46:usbmux daemon,,,:/home/usbmux:/bin/false
```

用户名　用户 ID　组ID　帳号的说明　主目录　登陆 Shell

加密后的口令

用passwd命令修改新增用户口令

```
root@toshiba:~# passwd kaseihi
Enter new UNIX password:
Retype new UNIX password:
passwd: password updated successfully
```

用id命令查看当前“我”的身份

```
kaseihi@toshiba:~$ id
uid=1000(kaseihi) gid=1000(kaseihi) groups=1000(kaseihi),24(cdrom),25(floppy),29(audio),30(dip
),44(video),46(plugdev),108(bluetooth),111(netdev)
```

变量

变量无需事先声明

变量值总是作为字符串存储，做数值计算时，自动转换为数字

创建或修改一个变量：变量名=变量值

显示变量的值：**echo $变量名**

删除变量：**unset 变量名**

导出环境变量：**export 变量名**

环境变量

set **env**

.bash_profile

.bash_history

.bashrc

/etc/profile：建立全系统通用的初始环境，用户每次登录时，第一个被执行

PATH PS1 PS2 HOME

source命令 .

cd

pwd

man

info

stdin(0)， stdout(1)， stderr(2)

ls | less
sort | grep aaa | tee bb | wc
>、 >>、 <、 <<

2>&1

set –o noclobber

>|

exec

文件名扩展

*

?

[,]　　[-]　　[!]

{,}

read命令从终端读入字符串放在指定的变量中

```
#!/bin/bash
echo -n “Enter name of file to delete:”
read file
echo “You want to delete $file ?”
echo “That was YOUR decision!”
```

test命令

test命令用来完成字符串比较，数值比较，文件测试和逻辑操作符

字符串比较:

= 比较两个字符串是否相等

test “1” = “2”

!= 比较两个字符串是否不等

test “1” = “1”

-n 检查字符串长度是否大于0

test –n “”

-z 检查字符串长度是否等于0

test -z “”

test命令

test命令用来完成字符串比较，数值比较，文件测试和逻辑操作符

数值比较:

-eq　比较两个数值是否相等

test 1 –eq 2

-ge　比较前者是否大于等于后者

-le　比较前者是否小于等于后者

-ne　比较两个数值是否不等

-gt　比较前者是否大于后者

-lt　比较前者是否小于后者

test命令

test命令用来完成字符串比较，数值比较，文件测试和逻辑操作符

文件测试比较:

-d 检查文件是否是一个目录

test –d .inputrc

-f 检查是否是个文件

-e 检查文件名是否存在

-r 检查对此文件或目录有“读”的权限

-s 检查文件长度是否大于 0

-w 检查对此文件或目录有“写”的权限

-x 检查对此文件或目录有“执行”的权限

test命令

test命令用来完成字符串比较，数值比较，文件测试和逻辑操作符

逻辑操作:

!	negate (NOT) a logical expression test ! 1 –lt 2
-a	logically AND two logical expressions test 1 –lt 2 –a 2 –lt 3
-o	logically OR two logical expressions test 1 –lt 2 –o 2 –lt 3

test命令

test命令，就是[命令，用]配对

[s1 = s2]	(true if s1 same as s2, else false)
[s1 != s2]	(true if s1 not same as s2, else false)
[s1]	(true if s1 is not empty, else false)
[-n s1]	(true if s1 has a length greater then 0, else false)
[-z s2]	(true if s2 has a length of 0, otherwise false)
[n1 -eq n2]	(true if n1 same as n2, else false)
[n1 -ge n2]	(true if n1greater then or equal to n2, else false)
[n1 -le n2]	(true if n1 less then or equal to n2, else false)
[n1 -ne n2]	(true if n1 is not same as n2, else false)
[n1 -gt n2]	(true if n1 greater then n2, else false)
[n1 -lt n2]	(true if n1 less then n2, else false)
[-d fname]	(true if fname is a directory, otherwise false)
[-f fname]	(true if fname is a file, otherwise false)
[-e fname]	(true if fname exists, otherwise false)
[-s fname]	(true if fname length is greater then 0, else false)
[-r fname]	(true if fname has the read permission, else false)
[-w fname]	(true if fname has the write permission, else false)
[-x fname]	(true if fname has the execute permission, else false)

expr：对表达式求值，并打印结果

expr 5 / 3

expr 5 % 2

if语句

```
if [ expression ];
then
        statements
elif [ expression ];
then
        statements
else
        statements
fi
```

```
#!/bin/bash
echo -n “Enter your login name: ”
read name
if [ “$name” = “$USER” ];
then
        echo “Hello, $name. How are you
today ?”
else
        echo “You are not $USER, so who
are you ?”
fi
```

```
#!/bin/bash
echo “Enter a path: ”; read x
if  cd $x; then
        echo “I am in $x and it contains”; ls
else
        echo “The directory $x does not
exist”;
        exit 1
fi
```

case语句

```
case $var in
val1)
        statements;;
val2)
        statements;;
*)
        statements;;
esac
```

for语句

```
for var in list
do
        statements
done
```

```
#!/bin/bash
for x in paper pencil pen
do
        echo “The value of variable x is: $x”
        sleep 1
done
```

偷懒的for语句

```
#!/bin/bash
for x
do
        echo “The value of variable x is: $x”
        sleep 1
done
```

while语句

```
while expression
do
        statements
done
```

until语句

```
until expression
do
        statements
done
```

函数

函数的声明要写在使用函数的语句之前

```
function 函数名()
{
        statements
}
```

特殊的shell变量

$#　传递给脚本或函数的参数个数
$0　脚本程序自身的名称（命令行名称）
$1　传给脚本或函数的第一个参数
$2　$3　$4
$?　上一条语句的返回值
$$　执行本脚本的进程的PID值
$*　将所有的参数作为一个字符串
$@　将所有的参数分解为包含若干字符串的数组