۳-۱۸ تنظیم اسامی مستعار

در لینوکس این امکان وجود دارد تا برای آسانی بیشتر، اسامی مستعاری را بجای aliasفرمان اصلی تعیین کنید .برای اضافه کردن اسامی مستعار باید از دستور استفاده کنید .به مثالهای زیر توجه کنید:

```
alias p="pwd; ls –CF"
alias rm="rm –I"
```

در مثال نخست حرف p دستور pwd را اجرا کرده و پس از آن دستور ls -CF اجرا خواهد شد که محتویات دایرکتوری جاری را چاپ خواهد کرد .در مثال دوم، دستور rm طوری تنظیم شده است تا فقط با گزینه i اجرا شود.

در صورتی که دستور alias را به تنهایی تایپ کنید، لیستی از اسامی مستعاری که تنظیم کرده اید نمایش داده میشود .توجه داشته باشید که اسامی مستعار در یک فایل پیکربندی ذخیره شده و با بستن پوسته فرمان از بین نمی روند.

۳-۱۹ دستورات پردازش فایل

تغییر مسیر پیغام های خطا :

سیستم عامل unix خطاهای احتمالی که در اثر اجرای یک فرمان بروز می کند را روی کانال استاندارد خطا یعنی standard error می نویسند. اگر بخواهیم پیغام های خطای حاصل از اجرای فرامین بجای نمایش روی خروجی, در یک فایل موردنظر ثبت و ذخیره شود باید از فرمان تغییر کانال خطای استاندارد استفاده کنیم. سیستم عامل کانالی با عدد 2 را برای خطا استفاده می کند. کانال صفر به ورودی استاندارد, کانال یک به خروجی استاندارد و کانال دو به خطا تخصیص یافته اند.

مثال :

```
ls  abcds  2 > err
```

نکته: برای دیدن محتوای این فایل از دستور cat استفاده می کنیم.

حذف فایل :

```
rm  -[fri] filename(s)
```

کاربرد دستور فوق برای حذف کردن فایلها می باشد. پس از اجرای فرمان پیغامی جهت تأیید عمل حذف پرسیده می شود و در صورت تأیید کاربر فایل حذف خواهد شد.

- i - برای پاک کردن فایل ها از کاربر تایید گرفته می شود
- f- فایل هایی که برای آنها اجازه w ندارد را بدون گرفتن تایید(force) پاک می کند.
- r- تمام فایل و زیر فهرست های یک دایرکتوری را به صورت بازگشتی (recursive) پاک می کند.

تمرین:

```
rm     f*
rm     [aAtT]*
rm     [a-f]?
```

دستور کپی cp :

این فرمان فایل یا پوشه مبدأ را در مسیر مقصد کپی می کند.

```
cp     -[fri]     source     destination
```

مثال :

```
cp     test1        apple/test2
cp      /dev/t*          /test
```

کاربرد سوئیچ های این فرمان بترتیب زیر می باشد:

- i - برای جایگزین کردن فایل ها از کاربر تایید گرفته می شود

- f- فایل هایی که برای آنها اجازه w ندارد را بدون گرفتن تایید جایگزین می کند.
- r- تمام فایل و زیر فهرست های یک دایرکتوری را به صورت بازگشتی (recursive) جایگزین می کند.

حذف فهرست ها :

برای حذف باید فهرست از قبل تهی باشد(مخفف removing directory). نام فهرست rmdir

اگر از سوئیچ p استفاده شود در صورتیکه دایرکتوری بالا هم خالی نباشد آن را نیز پاک می کند.

مثال :

```
rmdir     d2/d3    d1
rmdir      -p      d2/d3
```

دستور mv :

از این فرمان هم برای انتقال و هم برای تغییر نام فایلها و پوشه ها استفاده می شود.

```
mv    f1    f2    f3    dir1
```

تغییر مسیر f1 و f2 و f3 از مسیر جاری به dir1

```
mv     olddir     newdir
```

تغییر نام دایرکتوری از olddir به newdir

```
mv     file1     file2
```

تغییر نام فایل از file1 به file2

نکته: در صورتیکه نوع مبدأ و مقصد فرمان mv یکسان باشد (یعنی هر دو فایل باشند یا هر دو پوشه باشند) بشرط یکسان بودن مسیر موردنظر در فرمان, کاربرد تغییر نام برقرار می شود. اما اگر مبدأ فایل و مقصد پوشه باشد, یا مسیر بکار رفته در پارامترهای فرمان متفاوت باشند, کاربرد آن انتقال خواهد بود.

دستور od :

این فرمان نیز برای مشاهده محتویات فایلها بکار می رود, اما با بکارگیری آن همراه با سوئیچ های ذکر شده نتایج زیر حاصل خواهد شد:

- c - بایتهای محتویات فایل را بصورت حروف اسکی نشان می دهد.
- d - کلمات محتویات فایل را در مبنای ده نشان می دهد.
- o - کلمات محتویات فایل را در مبنای هشت (اکتال) نشان می دهد.
- x کلمات محتویات فایل را در مبنای شانزده نشان می دهد.

مثال :

```
od    -c    filename1
```

یافتن فایل ها :

با دستور find می توان موقعیت فایلهای خاصی را که نام و مشخصات دیگری از آنها را می دانیم, بیابیم. پارامترهای این فرمان بدین ترتیب هستند که می توان مسیر جستجو را برای فرمان find تعیین کرد که در صورت عدم ذکر آن, جستجو در کل حافظه سیستم انجام خواهد شد.

پارامتر type برای تعیین نوع فایل یا پوشه ای است که بدنبال آن می گردیم. می توان یکی از موارد زیر را بکار برد.

- b : فایل ویژه بلاکی
- p : فایل پایپ
- f : فایل معمولی
- l : فایل لینک
- d : پوشه

مثال :

```
find  /t1   -name  black   -type  d
find  -size  10240c
find   /t1   -size  10
```

دستور wc :

با اجرای این فرمان می توان تعداد حروف, کلمات و خطوط یک یا چند فایل را مشاهده نمود. سوئیچ های این فرمان بترتیب کاربردهای زیر را دارند:

- -c شمارش تعداد حروف
- -w شمارش تعداد کلمات
- -l شمارش تعداد خطوط

مثال :

```
wc    filename1   filename2
```

دستور grep :

```
grep   -[vcln] pattern  files
```

این دستور خطوطی از محتوای فایل که شامل عبارت موردنظر هستند را پیدا کرده و روی صفحه نمایش می دهد. چنانچه الگوی موردنظر را در بیش از یک فایل جستجو کنیم خطوط پیدا شده را با ذکر نام فایل روی صفحه نمایش می دهد. پارامترهای این فرمان عبارتند از:

- -v این پارامتر خطوطی را نشان می دهد که الگوی موردنظر در آنها وجود نداشته باشد.
- -c این پارامتر تعداد دفعات تکرار یک رشته را در یک پرونده نشان می دهد.
- -l این پارامتر باعث می شود فقط اسامی فایلهایی که حاوی رشته هستند نشان داده شود.
- -n شماره خط را الگو را قبل از نمایش آن روی صفحه می نویسد.

مثال :

```
grep  -v  “the”   .bash
```

pipes:

در بسیاری از موارد لازم است که از خروجی یک دستور بعنوان ورودی دستور دیگر استفاده نمائیم. با استفاده از علامت " | " که pipe نامیده می شود کاربر به سیستم عامل می گوید که خروجی یک دستور بعنوان ورودی دستور بعدی در نظر گرفته شود و بدین ترتیب چند دستور با یکدیگر تلفیق می شوند.

برای ایجاد یک سه راهی در میان فرامین Pipe می توان از فرمان tee استفاده کرد.این فرمان بدون هیچگونه تغییر ورودی استاندارد خود را به خروجی استاندارد منتقل کرده و همزمان یک کپی از داده های ورودی را در فایلهای اختصاص داده شده از سوی کاربر ایجاد می کند.

مثال :

```
command1 | tee file |command2
```

cat apple | grep "canvas" | tee mine | wc -l

در این مثال تعداد خطوطی از فایل apple که حاوی رشته canvas باشند بعنوان خروجی نشان داده شده, علاوه بر آن یک کپی از خروجی فرمان Grep در فایل mine نیز ذخیره می گردد.

دز توضیح فرمان فوق می توان گفت ابتدا فرمان cat apple محتوای فایل apple را همانگونه که قبلاً آموختید نمایش می دهد. سپس فرمان grep "canvas" روی خروجی فرمان قبل اثر کرده و در خطوطی که حاصل از اجرای فرمان قبلی هستند بدنبال آنهایی می گردد که حاوی رشته canvas باشند. بنابراین خروجی این فرمان خطوطی از فایل apple است که رشته canvas در آنها وجود داشته باشد. فرمان نهایی یعنی wc -l کارش شمارش تعداد خطوط یک فایل است. ورودی این فرمان, خروجی فرمان قبل است. پس تعداد خطوط فرمان قبل را نشان می دهد و درنتیجه تعداد خطوط فایل apple که دارای رشته canvas هستند در خروجی نمایش داده خواهد شد.

فرمان tee همانگونه که ذکر کردیم یک سه راهی در میان فرامین Pipe ایجاد کرده و خروجی حاصل از فرمانهای پیش از خود را در فایل مشخص شده در فرمان tee ذخیره می نماید.

۳-۲۰ حساب کاربری

برای ساختن یک user از فرمان **useradd** استفاده میکنیم

useradd Roohollah

با توجه به اینکه لینوکس یک سیستم عامل فوق العاده secure است برای user نیز باید یک پسورد داشته باشیم.از فرمان **passwd** همراه با نام user میتوانیم پسورد را تعیین کنیم

passwd Roohollah

سپس دودفعه باید پسورد مورد نظر را وارد کنید.توجه کنید که با توجه به security بالا لینوکس پسورد حتی با ستاره هم نشان داده نمی شود

اگر بخواهیم یک اطلاعات اضافی مثل نام و نام خانوادگی و یا شماره تلفن به یک user اختصاص دهیم از پارامتر **c**- میتوانیم استفاده کنیم

useradd -c " Roohollah Pishbahar" Roohollah

اگر بخواهیم تغییری در user که درست کرده ایم بدهیم از فرمان **usermod** که به معنی modify است استفاده کنیم

usermod -c " AsreDanesh web site's" Roohollah

پروفایل هر user جدیدی که درست میکنیم در دایرکتوری home/ ذخیره میشود.user ها ساخته شده همرا با اسم user, ,شماره UID-User ID ,GID-Group ID,home directory و shell که زمانی که کاربر وارد میشود در گردش میفتد را میتوانیم در شاخه زیر مشاهده کنیم

cat /etc/passwd

با فرمان زیر میتوانیم user مورد نظر را حذف کنیم

userdel Roohollah

توجه کنید با اجرای فرمان بالا حساب کاربری کاربر حذف میشود اما اگر در دایرکتوری home/ برویم اسم کاربر را مشاهد می کنیم.به منظور حذف کامل حساب کاربر میتوانیم از فرمان زیر استفاده کنیم

userdel -**r** Roohollah

با فرمان زیر می توانیم یک گروه برای کاربران درست کنیم

groupadd -r project

از فرمان زیر به منظور اضافه کردن کاربر به گروه استفاده میکنیم

usermod -G project Roohollah

در مسیر زیر کاربرانی که عضو گروه ما هستند قابل دیدن میباشند

cat /etc/group

با فرمان پایین میتوانیم یک کاربر را عضو دو گروه مختلف کنیم.برای این منظور یک گروه جدید میسازیم سپس کاربر را عضو هر دو گروه می کنیم

groupadd -r proj
usermod -G proj,project Roohollah

از فرمان زیر در جهت اینکه کاربر عضو کدام گروه است استفاده میشود

groups Roohollah

برای عوض کردن مالک یک فایل در گروه از فرمان زیر استفاده میکنیم.توجه کنید اسم فایل ۱۲۳ و اسم کاربر Roohollah و اسم گروه project در نظر گرفته شده است

chown Roohollah.project 123

اگر بخواهیم یک کاربر عادی را مدیر یک گروه کنیم از فرمان زیر استفاده میکنیم

gpasswd -**A** Roohollah project

حال کاربری که مدیر فایل شده قابلیت این را دارد که کاربر جدید به گروه اضافه و یا از گروه خارج کند

مدیر گروه جدید از فرمان زیر به منظور اضافه کردن کاربر به گروه استفاده میکند

gpasswd -a ali project

مدیر گروه جدید از فرمان زیر به منظور حذف کردن کاربر از گروه استفاده میکند

gpasswd -d ali project

۳-۲۰-۱ دستور users,who و w

بوسیله فرمان **who** نام کاربران وارد شده در سیستم نشان داده میشود

who
who am i

پارامترهای دستور Who

b- : آخرین باری که سیستم بوت شده کی بوده ؟

p- : چه پروسس هایی الان فعال هستند ؟

u- : زمان هایی که یوزر خاص مورد نظر ما لوگین می شده کی بوده ؟

با استفاده از دستور **w** اطلاعات بیشتری در مورد کاربران نشان داده می شود

w

```
19:51:32  up  6:18,  2 users,  load average: 0.44, 0.57, 0.87
USER     TTY      FROM             LOGIN@   IDLE   JCPU   PCPU  WHAT
Roohollah tty7     :0               13:33    6:18m  34:57  0.42s gnome-session
Roohollah pts/0    :0.0             19:37    0.00s  0.32s  0.00s w
```

توضیح سطر اول: مدت زمانی که کامپیوتر روشن شده-تعداد کاربرها-بارگزاری متوسط توسط پردازنده

توضیح سطر دوم: کاربرانی که وارد سیستم شده اند- مکان ورود- زمان ورود- مدت زمان ورود – میزان اشغال سی پی یو – برنامه در حال اجرا

تفاوت JCPU و PCPU:

JCPU: زمان کل پردازش هایی که به tty ضمیمه شدند (شامل job هایی که در گذشته در پس زمینه انجام میشدن نمیشه ولی شامل اونایی که الآن دارن در پس زمینه لینوکس انجام میشن هست).

PCPU: زمان استفاده شده بوسیله پروسس فعلی که در ستون what از خروجی دستور w نام پروسس نمایش داده می شود

با استفاده از دستور **users** لیست کاربرانی که مجوز ورود دارند نمایش داده می شود

users

با استفاده از دستور **finger** اطلاعات کاملی راجع به یک کاربر نمایش داده می شود

finger نام کاربر

۳-۲۱ مدیریت پردازش

یک پروسس, یک برنامه در حال اجراست. هر پروسس بوسیله یک شماره واحد شناخته می شود. به این شماره PID یا شماره مشخصه پروسس گفته می شود. پروسسهای شماره صفر و یک ویژه سیستم می باشند. پروسس شماره صفر Kernel سیستم عامل و پروسس شماره یک, پروسس Init نام دارد. این پروسس وظیفه برپاسازی ساختار پروسسها را بر عهده دارد. در سیستم عامل unix تمام پروسسها توسط یک پروسس دیگر ایجاد می شوند که به آن پروسس والد (parent) گویند.

دستور PS :

این فرمان PID و اطلاعات دیگری در مورد پروسسهای در حال اجرا را نمایش می دهد. اگر هیچ سوئیچی را در سطر فرمان به PS اختصاص ندهید اطلاعات زیر را مشاهده خواهید کرد:

- PID شماره مشخصه پروسس.
- tty نام ترمینالی که پروسس را اجرا کرده است.
- Time زمان مصرف شده برای اجرای فرمان.
- Command نام فرمان

اگر فرمان PS همراه با سوئیچ -p بکار گرفته شود, می توان اطلاعات پروسسهای با شماره خاص را مشاهده کرد.

مثال: ps -p 100,107,267

- اگر فرمان PS همراه با سوئیچ -f بکار گرفته شود, یک لیست کامل از اطلاعات مختلف درباره پروسسهای فعال را نمایش می دهد. این اطلاعات شامل موارد زیر است:
  - نام کاربری که پروسس را فعال کرده (UID)
  - شماره مشخصه پروسس (PID)
  - شماره پروسس والد (PPID)
  - شماره process utilization
  - ساعت شروع پروسس (stime)
  - ترمینالی که پروسس روی آن فعال شده است.
  - مدت زمانی که اجرای پروسس طول کشیده است. (Time)
  - دستوری که اجرا شده. (command)

بوسیله فرمان **tty** میتوانیم تشخیص دهیم کاربر در کدام ترمینال در حال فعالیت میباشد

tty

۳-۲۲ کار با DVD,CD,Flopy,...

محتویات CD,dvd و یا flopy disk در دایرکتوری media موجود می باشد و برای دسترسی به آن مـی بایـست آن را mount کنیم

مثلا برای دسترسی به CD ROM می بایست دستور زیر را وارد کنیم

mount /media/cdrom

در صورتی که بخواهیم CD را از داخل CD ROM خارج کنیم به دلیل mount بودن این عمل میسر نمی باشد و ابتدا باید CD را un mount کنیم

بدین منظوراز فرمان زیر استفاده می کنیم

umount /media/cdrom

برای خارج کردن CD از CD ROM از فرمان **eject** استفاده می کنیم

eject

۳-۲۳ تنظیماتIP در لینوکس

در هرشبکه بطورمعمول نخستین لایه یعنی سخت افزار از یک کارت شبکه یا اترنت تشکیل شده و برای اینکه این کارت بعنوان یک رابط درمحیط شبکه بکارگرفته شود بایستی اولا آدرس واحدی تحت عنوان IP address به آن اختصاص یافته و ثانیا"بسته ها یا packet های اطلاعاتی برای رسیدن به این رابط مسیر دهی شوند که به این عمل routing گفته میشود.

در اغلب توزیعهای لینوکس راههای خاصی برای انجام تنظیمات رابط یا کارت شبکه گنجانده شده که در اینجا روش ساده استفاده از فرامین متنی در محیط shell بررسی می شود :

باید دانست که هسته یا کرنل سیستم عامل لینوکس بطورپیش فرض از یک رابط مجازی یا interface (lo loopback) بعنوان یک ابزار مجازی برای ارتباط با خود سیستم استفاده می کند .

این ابزار مجازی بطور معمول IP 127.0.0.1 و نام localhost را برمی گزیند و در هنگام بوت سیستم بطور ازپیش تعریف شده فعال میگردد .

قسمت اصلی سخت افزار شبکه یک رایانه کارت شبکه (**NIC(N**etwork **I**nterface **C**ard) می باشد که در محیط گنو/لینوکس معمولا با اسامی eth0 یا eth1 و ... بسته به تعداد کارتهای شبکه متصل به سیستم آدرس دهی و نامگذاری میشود.

اگر بر روی سیستم ما دو کارت شبکه موجود بود و تمایل داشتیم یکی از آنها را غیرفعال و یا فعال کنیم از فرمانهای زیر استفاده می کنیم

برای غیر فعال کردن از

ifdown eth0

برای فعال کردن از

ifup eth1

برای تنظیم IP یک کارت شبکه بافرض اینکه این کارت اولین و تنها کارت شبکه نصب شده در سیستم است تنها چیزی که لازم داریم برنامه (interface configure)**ifconfig** می باشد و می بایست بعنوان مثال از دستور زیر در محیط shell با مجوز کاربر ریشه استفاده کنیم :

ifconfig eth0 192.168.3.8 broadcast 192.168.3.255 netmask 255.255.255.0

همانطوریکه ملاحظه می شود در اینجا یک IP کلاس C شامل آی پی اختصاص یافته به کارت شبکه ۱۹۲,۱۶۸,۳,۸ و یک پوشانه زیر شبکه(subnetmask) کلاس c شامل ۲۵۵,۲۵۵,۲۵۵,۰ به کارت شبکه ما eth0 اختصاص می یابد . و آی پی broadcast نیز شامل IP خاصی است که کلیه کامپیوترهای شبکه به آن پاسخ می دهند و معمولا آخرین شماره آی پی یک کلاس شبکه می باشد .

در ادامه برای بررسی صحت عملیات انجام شده و عملکرد کارت شبکه از دستور **ifconfig** به تنهائی استفاده کرده و پاسخ سیستم را که چیزی مشابه زیر خواهد بود مشاهده می کنیم :

ifconfig

```
eth0      Link encap:Ethernet  HWaddr 00:21:70:b2:1b:5f
          inet addr:192.168.0.1  Bcast:192.168.0.255  Mask:255.255.255.0
          UP BROADCAST MULTICAST  MTU:1500  Metric:1
          RX packets:0 errors:0 dropped:0 overruns:0 frame:0
          TX packets:0 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:0 (0.0 B)  TX bytes:0 (0.0 B)
          Interrupt:30 Base address:0x2000
lo        Link encap:Local Loopback
          inet addr:127.0.0.1  Mask:255.0.0.0
          inet6 addr: ::1/128 Scope:Host
          UP LOOPBACK RUNNING  MTU:16436  Metric:1
          RX packets:8 errors:0 dropped:0 overruns:0 frame:0
          TX packets:8 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:480 (480.0 B)  TX bytes:480 (480.0 B)
```

از فرمان ifconfig برای دیدن و تغییر IP address کارت شبکه که به سیستم ما تعلق گرفته است استفاده می کنیم .دراین گزارش حتی آدرس سخت افزاری کارت شبکه eth0 هم مشخص شده 00:21:70:b2:1b:5f که در نوع خود با توجه با سادگی دستورات داده شده نتایج جالبی به نظر میرسد .

برای تغییر آدرس یک کارت شبکه:

Ifconfig eth0 192.168.0.1 netmask 255.255.255.0

۳-۲۴ پیکربندی پوسته فرمان

برای اینکه بتوانید بطور موثرتری از پوسته فرمان خود استفاده کنید، میتوانید آنرا بنا به خواسته خود تنظیم کنید. برای این منظور باید فایلهای پیکربندی پوسته فرمان خود را ویرایش کنید.

تعدادی فایل پیکربندی وجود دارد که نحوه رفتار پوسته فرمان شما را تعیین میکند. برخی از این فایلها برای تمام کاربران و پوسته ها مشترک بوده و برخی مخصوص یک کاربر خاص هستند. فایلهای پیکربندی زیر فایلهایی هستند که هر کاربر پوسته فرمان در لینوکس از آنها استفاده میکند :

- etc/profile : این فایل اطلاعات محیط کاربری هر کاربر را ذخیره میکند. این فایل هنگامی اجرا میشود که شما به سیستم وارد شده و پوسته فرمان آغاز به کار میکند. این فایل مقادیر پیش گزیده مسیر، شکل اعلان فرمان، حداکثر تعداد فایلی که شما میتوانید ایجاد کنید و مجوز های پیش گزیده برای فایلهایی که ایجاد میکنید را تعیین میکند. همچنین این فایل متغییر های محیطی مانند محل صندوق پستی و اندازه فایلهای تاریخچه را تنظیم میکند.
- etc/bashrc : این فایل برای هر کاربری که پوسته bash را اجرامیکند، اجرا میشود. این فایل حالت اعلان فرمان را تنظیم میکند. مقادیر این فایل میتواند توسط فایل bashrc که در دایرکتوری خانگی هر کاربر وجود دارد، تحت تاثیر قرار گیرد.
- bashrc./~ : این فایل حاوی اطلاعات مربوط به bash هر کاربر میباشد. این فایل هنگامی خوانده میشود که به سیستم وارد میشود و هر گاه که یک پوسته جدید باز میکنید. اینجا بهترین مکان برای ذخیره متغییرهای محیطی و فرمانهای مستعار خاص خودتان است.
- bash_profile./~ : این فایل برای وارد کردن اطلاعات خاصی که هر کاربر در استفاده از پوسته بکار میبرد میباشد. این فایل تنها یکبار اجرا میشود. هنگامی که کاربر به سیستم وارد میشود. این فایل تعدادی از متغییرهای محیطی را مقدار دهی کرده و فایل bashrc مربوط به کاربر را اجرا میکند.
- bash_logout./~ : این فایل هر گاه که شما از سیستم خارج میشوید اجرا میشود. این فایل فقط صفحه نمایش را پاک میکند.

برای تغییر فایلهای etc/profile و etc/bashrc باید با کاربر ریشه وارد سیستم شده باشید. هر کاربر میتواند اطلاعات موجود در فایلهای bash_profile، bashrc و bash_logout موجود در دایرکتوری های خود را تغییر دهد.

۳-۲۵ نوشتن برنامه در محیط لینوکس

نوشتن برنامه در سیستم عامل لینوکس همانند سیستم عامل ویندوز می باشد با این تفاوت که سیستم عامل لینوکس خود حاوی کامپایلر C++ می باشد.

برای نوشتن برنامه باید ابتدا در یک محیط برنامه را ویرایش کرد و بعد با پسوند CPP، آن را ذخیره کرد. محیطهای متفاوتی در لینوکس برای نوشتن برنامه موجود می باشد .یکی از این محیطها gedit می باشد. برای دستیابی به این محیط در ترمینال(Terminal) کلمه **gedit** را تایپ کرده و کلید Enter را فشار دهید. یکی از قابلیتهای این محیط تنظیم محیط با زبان مورد نظر که در اینجا C++ می باشد. بعد از این تنظیم کامپایلر کلمات کلیدی و رزرو شده را تشخیص داده و به نوشتن برنامه کمک خواهد کرد اگر سیستم عامل لینوکس شما حاوی محیط گرافیکی نباشد می توانید بجای gedit از محیط Vi استفاده نمائید.

برای دستیابی به این محیط از دستور Vi در ترمینال استفاده نمائید.

بعد از آن که برنامه را با پسوند CPP نمودیم .باید آن را با دستور زیر کامپایل نماییم.

g++ اسم برنامه.CPP

مثلا اگر برنامه به نام exam م یباشد بصورت زیر کامپایل می شود.

G++ exam.CPP

یک بر نامه خیلی ساده به طور مثال برنامه خوش آمد گویی را تایپ کنید . مثال زیر یک برنامه بسیار ساده می باشد .

```
#include

Int main()

{

        Cout << "welcome to c++";

        Returne 0;

}
```

سپس این برنامه را با نام دلخواه و با پسوند cpp ذخیره کرده و از محیط ادیتور خارج شوید . مرحله کد نویسی تمام شده حالا باید برای برنامه نوشته شده را کامپایل و اجرا کنید. برای این کار یک کنسول یا ترمینال باز کنید . برای کامپایل کردن برنامه از دستور g++ استفاده می کنیم.

```
g++   -o t2   test1.cpp
```

دستور مقابل از چند ستون تشکیل شده است . ستون اول g++ مربوط به دستور کامپایل کردن می باشد. در ستون دوم سوئیچ o- برای ساختن فایل خروجی یا اجرای بعد از کامپایل کردن مورد استفاده قرار گرفته است. درستون سوم t2 اسم فایلی است که بعد از کامپایل کردن ایجاد می شود ویک فایل اجرایی میباشد . ودر ستون آخر test1.cpp اسم فایلی است که شما برنامه c++ را در اون ذخیره کردید بعد از اجرای دستور بالا ابتدا برنامه شما کامپیال شده و سپس یک فایل با اسم t2 ساخته میشود .

نکته: اگر در برنامه شما خطا وجود داشته باشد فایلی برای اجرا ساخته نمیشود و در مرحله کامپایل شماره خطهای دارای خطا را نمایش می دهد . اگر برنامه بدون خطا اجرا شود کامپایل به صورت زیر در کنسول اجرا شده و یک فایل ساخته می شود.

```
G++ -o t2 test1.cpp

In file included from /usr/lib/gcc/i386-redhat-

Linux/3.4.2/../../../../include/c++/3.4.2/backward/iostream.h:31;

From test1.cpp:1:

/usr/lib/gcc/1386-redhatg-

Linux/3.4.2/../../../../include/c++/3.4.2/backward-warning.h:32:2:warning:#warning this file includes at last one deprecated or antiquated header.please

 Consider using one of the 32 headers found in section 17.4.12 of the c++ standard.

Examples include substituting the header for the header for c++ includes ,or instead of the

Deprecated header. To disable this warning use-Wno-deprecated.

Test1.cpp:6:2: warning:no newline at end of file
```

حالا می توانید فایل ساخته شده را اجرا کنید و نتیجه کار را ببینید.

```
./t2

Welcome to c++
```

ستون اول **./t2** برنامه را به اجرا در می آورد.

مشاهده می کنید که بعد از اجرای برنامه پیغام خوش آمود گویی welcome to c++ را چاپ می کند .

۳-۲۲ سرانجام پنگوئن محبوب توروالدز چه خواهد شد؟

"ابتدا تو را نادیده میگیرند، سپس مسخره ات میکنند و بعد با تو می جنگند. ولی در نهایت پیروزی از آن توست"

«گاندی»

۳-۲۳ دستور کار:

۱. سعی کنید که به شاخه‌های /root و/home/user2 وارد شوید و محتویات آنها را مشاهده کنید. هر دفعه با چه پیامی مواجه می‌شوید؟ همین کار را در مورد شاخه‌های /bin و /lib انجام دهید.

۲. در محیط وارد شده و با امتحان کردن دستورات و برنامه‌های زیر با کار هریک آشنا شوید:

--help , help , info, man , pwd , mv, rm , mkdir , cd , cp , ls, who, chmod , grep tar , mc , mcedit , wget , make , gcc ,**ifconfig**

۳. تفاوت دستورهای help– و help و info و man را توضیح دهید.

۴. فرمان ls دارای گزینه‌هایی است که برخی از آنها عبارتند از : a- و l– . از راهنمای سیستم عملکرد هرکدام از گزینه‌های زیر را بیابید و آنها را آزمایش کنید. سپس نتیجه هریک را در کاربرگ خود بیاورید.

۵. با فرمان mkdir یک شاخه جدید ایجاد کنید. توسط پارامتر p– این فرمان سعی کنید شاخه و زیرشاخه جدیدی را همزمان ایجاد کنید. دستور را در کاربرگ یادداشت نمایید.

۶. دستور who|grep john? چه عملی را انجام می‌دهد؟

۷. تفاوت دستور who|sort و who>sort در چیست؟

۸. تفاوت make و make depend را توضیح دهید.

۹. عملکرد تابع getpid را از صفحات راهنما به وسیله دستور man بیابید. با استفاده از این دستور در پرونده pid.c برنامه‌ای بنویسید که شماره PID خود را چاپ کند. این برنامه را با استفاده از gcc ترجمه و به دفعات اجرا کنید. چه نتیجه‌ای می‌گیرید؟

۳-۲۴ سوالات

۱. لینوکس و ویندوز را باهم مقایسه کنید(نقاط ضعف و قوت لینوکس نسبت به ویندوز؟)

۲. آیا می توان در محیط لینوکس با محیط دات نت برنامه نویسی و سپس آن را اجرا کرد؟پاسخ خود را توضیح دهید.

۳-۲۵ منابع:

www.technotux.org
www.farsilinux.org
www.linuxdoc.org
www.irantux.com
www.ospdev.net
www.gnome.org
www.IRITN.com
tafazoli.iut.ac.ir
www.atcce.com
www.linux.com
www.linux.org
www.subnet.ir
www.suma.ir

تحقیق

چگونه می توان بخشی از سیستم عامل لینوکس را فارسی نمود؟ مزیت ها و معایب این عمل را بیان کنید.