مترجم:سيد حسن عطارزاده نياكي

shaniaki@yahoo.com

فصل 5- هدايت ورودي/خروجي ها

در اين فصل در خصوص راهكارهاي قدرتمند يونيكس در هدايت ورودي، خروجي و خطاها صحبت مي شود. مباحث اين فصل شامل موارد زير مي باشد:

- ورودي، خروجي و خطاهاي استاندارد
- عملگرهاي هدايت
- چگونگي استفاده خروجي يك دستور به عنوان ورودي دستوري ديگر
- هدايت ورودي
- اداره كردن پيام هاي خطاي استاندارد
- تركيب هدايت جريانات ورودي، خروجي و خطا
- فيلترهاي خروجي

5.1: ورودي و خروجي هاي استاندارد چه هستند؟

اكثر دستورات لينوكسي ورودي را به عنوان فايل ورودي يا ديگر خصوصيات دستور مي خوانند، و نتيجه كار را در خروجي مي نويسند.به طور پيش فرض ورودي از صفحه كليد خوانده شده و خروجي روي صفحه به نمايش در مي آيد. صفحه كليد شما به عنوان دستگاه «ورودي استاندارد» (stdin) و صفحه نمايش شما يا هر پنجره پايانه ديگر به عنوان دستگاه «خروجي استاندارد» (stdout) خوانده مي شود.

با اين حال، از آن جا كه لينوكس يك سيستم انعطاف پذير است، لزومي ندارد كه در آن حتما اين تنظيمات پيش فرض به كار برده شود. به عنوان نمونه خروجي استاندارد در سروري كه شديدا تحت كنترل و برسي است ممكن است يك پرينتر باشد.

5.1.1عملگرهاي هدايت

5.1.1.1هدايت خروجي با > و |

گاهي اوقات مي خواهيد خروجي يك دستور را در يك فايل بنويسيد يا اينكه از آن به عنوان ورودي يك دستور ديگر استفاده كنيد. به اين كار هدايت خروجي مي گويند. هدايت با استفاده از عملگرهاي «>» (علامت بزرگتر)، يا «|» (پايپ) انجام مي گيرد كه خروجي استاندارد يك دستور را به ورودي استاندارد يك دستور ديگر مي فرستد.

همانطور كه قبلا ديديم دستور **cat** چند فايل را به هم مي چسباند و آن ها را با هم در خروجي استاندارد نمايش مي دهد. با هدايت خروجي به يك فايل، فايل نامبرده ساخته مي شود يا روي فايل هم نام قبلي نوشته مي شود، پس مواظب باشيد:

```
nancy:~> cat test1 some words

nancy:~> cat test2 some other words

nancy:~>cat test1 test2 > test3

nancy:~> cat test3 some words some other words
```

هدايت «هيچ چيز» به يك فايل موجود معادل خالي كردن آن فايل مي باشد:

```
nancy:~> ls -l list
```

```
rw-rw-r-- 1 nancy nancy 117 Apr 2 18:09 list-

nancy:~> > list

nancy:~> ls -l list -rw-rw-r-- 1

nancy nancy 0 Apr 4 12:01 list
```

اين عمل كوتاه كردن(truncating) ناميده مي شود.

هدايت هيچ چيز به فايلي كه وجود ندارد موجب ساخته شدن يك فايل خالي مي گردد:

```
nancy:~> ls -l newlist

ls: newlist: No such file or directory

nancy:~> > newlist

nancy:~> ls -l newlist

rw-rw-r-- 1 nancy nancy 0 Apr 4 12:05 newlist-
```

فصل 7 شامل مثال هاي بيشتري در مورد اين نوع هدايت مي باشد.

چند مثال در مورد پايپ كردن دستورات:

براي يافتن كلمه اي در يك متن، تمام خطوط حاوي «pattern1» را نمايش بده و از نمايش تمام خطوط شامل «pattern2» خودداري كن:

```
grep pattern1 file | grep -v pattern2
```

براي نمايش صفحه به صفحه خروجي ليست محتويات يك دايركتوري:

```
ls -la | less
```

براي جستجوي يك فايل در يك دايركتوري:

```
ls -l | grep part_of_file_name
```

5.1.1.2- هدايت ورودي

در مواردي ديگر شما مي خواهيد فايلي را به عنوان ورودي به دستوري بدهيد كه به طور عادي ورودي به صورت فايل قبول نمي كند. اين هدايت ورودي توسط عملگر «<» (علامت كوچكتر) انجام مي شود.

در زير مثالي از ارسال يك فايل به يك نفر توسط هدايت ورودي مشاهده مي گردد:

```
andy:~> mail mike@somewhere.org < to_do
```

اگر كاربر *mike* روي سيستم وجود دارد نيازي به تايپ كامل آدرس نيست. ولي اگر مي خواهيد به شخصي روي اينترنت دسترسي داشته باشيد آدرس كامل را به عنوان آرگومان ورودي وارد نماييد.

اين دستور كمي ناخوانا تر از دستور مبتدي **cat file | mail someone** مي باشد ولي بدون شك يك راه صحيح تر استفاده از امكانات موجود مي باشد.

5.1.1.3 تركيب هدايت ها

دستور زير هدايت ورودي و خروجي را تركيب مي كند. ابتدا اشكالات املايي فايل text.txt بررسي مي شود و سپس خروجي به يك فايل log خطاها هدايت مي شود:

```
aspell < text.txt > error.log
```

ممكن است از نگارش زير براي كار فوق استفاده كنيد(به صفحات راهنما رجوع كنيد):

```
aspell -H list < file.txt | sort -u
```

اين هم نوعي از هدايت ورودي و خروجي ها مي باشد دستور زير تمام دستوراتي را كه با استفاده از less مي توانيد با آن يك فايل ديگر را بررسي كنيد ليست مي كند:

```
mike:~> less --help | grep -i examine
 .e [file] Examine a new file:
 .n * Examine the (N-th) next file from the command line:
 .p * Examine the (N-th) previous file from the command line:
 x * Examine the first (or N-th) file from the command line:
```

گزينه –i به منظور جستجوي غير حساس به حروف بزرگ و كوچك استفاده شده است. به ياد داشته باشيد كه يونيكس به شدت به حروف بزرگ و كوچك حساس است.

اگر مي خواهيد خروجي اين دستور را براي مراجعات بعدي در اختيار داشته باشيد، خروجي آن را به يك فايل هدايت كنيد:

```
mike:~> less --help | grep -i examine > examine-files-in-less
mike:~> cat examine-files-in-less
 .e [file] Examine a new file:
 .n * Examine the (N-th) next file from the command line:
 .p * Examine the (N-th) previous file from the command line:
 .x * Examine the first (or N-th) file from the command line:
```

خروجي يك دستور به طور مجازي مي تواند بارها به ورودي يك دستور ديگرپايپ شود، اين عمل تا زماني كه اين دستور به طور معمولي از ورودي استاندارد مي خواند و در خروجي استاندارد مي نويسد مي تواند انجام شود. گاهي اوقات دستورات

اين كار را نمي كنند ولي ممكن است اين دستورات گزينه هايي داشته باشند كه آن ها را وادار به اين كند كه با تعاريف استاندارد كار كنند. بنابراين در صورت برخورد به خطا صفحات راهنما (man و info)ي دستورات را مطالعه كنيد.

روي فايل هاي موجود ننويسيد!

مواظب باشيدكه هنگام هدايت خروجي، روي فايل ها(ي مهم) بازنويسي نكنيد. در بسياري از shell ها از جمله Bash امكاني درون ساخته وجود دارد كه شما را از اين خطر در امان نگه مي دارد:noclobber. صفحاتinfo را براي اطلاعات بيشتر مطالعه كنيد. در bash مي توانيد با اضافه كردن دستور set -o noclobber در فايل پيكره بندي .bashrc تان مانع بازنويسي اتفاقي فايل هايتان شويد.

5.1.1.4عملگر >>

به جاي باز نويسي اطلاعات فايلتان مي توانيد ازدو علامت بزرگتر متوالي براي اضافه كردن به محتويات فايلتان استفاده كنيد:

```
mike:~> date >> wishlist

mike:~> cat wishlist

more money

less work

Thu Feb 28 20:23:07 CET 2002
```

دستور date در حالت عادي خط آخر را نمايش مي دهد. اما اكنون خروجي آن به آخر wishlist اضافه شده است.

5.1.2ويژگي هاي پيشرفته هدايت

5.1.2.1.: استفاده از توصيف گر هاي فايل ها

سه نوع ورودي/خروجي، با عددي مخصوص به خود به نام توصيف گر هاي فايل موجودند:

- ورودي استاندارد:0
- خروجي استاندارد:1
- خطاي استاندارد:2

در توصيف گر هاي زير اگر از عدد توصيف گر فايل صرفنظر شده باشد و كاراكتر اول عملگر هدايت < باشد، هدايت به سمت ورودي استاندارد خواهد بود(توصيف گر فايل 0). اگراولين كاراكتر عملگر هدايت > باشد هدايت به سمت خروجي استاندارد مي باشد(توصيفگر فايل 1).

چند مثال عملي اين موضوع را روشن تر مي كند:

```
ls > dirlist 2>&1
```

هر دوي خروجي استاندارد و خطاي استاندارد را به dirlist هدايت مي كند، در حالي كه دستور:

```
ls 2>&1 > dirlist
```

فقط خروجي استاندارد را به dirlist هدايت مي كند. اين يك امكان مناسب براي برنامه نويسان است.

تمام اين موضوعات به طور تفصيلي در صفحات info مورد بررسي قرار گرفته است.

5.1.2.2: چند مثال

5.1.2.2.1:تحليل خطاها

اگر process شما خطاهاي زيادي توليد مي كند اين يك راه براي برسسي كامل آن است

```
command 2>&1 | less
```

اين كار معمولا در هنگام ساخت نرم افزاري جديد با make صورت مي گيرد. مثلا:

```
--andy:~/newsoft> make all 2>&1 | less --output ommitted--
```

5.1.2.2.2:جداسازي خروجي استاندارد از خطاي استاندارد

چنين كاري معمولا توسط برنامه نويسيان صورت مي پذيرد و در آن خروجي را به يك پايانه نمايشي مي فرستند و خطا را به يكي ديگر،ابتدا با استفاده از tty ببينيد كه از كدام شبه ترمينال استفاده مي كنيد:

```
andy:~/newsoft> make all 2> /dev/pts/7
```

5.1.3:فيلترها

وقتي برنامه اي عملياتي را روي ورودي انجام دهد و حاصل را در خروجي بنويسد يك فيلتر ناميده مي شود. يكي از معمول ترين استفاده ها از فيلتر ها بازسازي خروجي است. در زير چند فيلتر مهم را بررسي مي كنيم.

5.1.3.1: كمي بيشتر درباره grep

همانطور كه در بخش 3.3.3.4 ديديم، grep خروجي را خط به خط در جستجوي يك الگو بررسي مي كند. تمام خطوط حاوي الگوي مزبور در خروجي استاندارد نوشته مي شوند. اين عملكرد مي تواند توسط گزينه –v معكوس شود.

چند مثال: فرض كنيد مي خواهيم ببينيم چند فايل در يك دايركتوري در فوريه دستكاري شده اند:

```
jenny:~> ls -la | grep Feb
```

دستور grep، مانند بسياري از دستورات، حساس به حروف بزرگ و كوچك است. براي اينكه آن را نسبت به اين موضوع غير حساس كنيد از گزينه –i استفاده كنيد. ويژگي هاي اضافي GNU بسياري نيز موجود است. مثلا --colour كه در هايلايت كردن عبارات طولاني جستجو مفيد است، يا after-context--كه تعداد خطوط را پس از آخرين خط مي نويسد. شما مي توانيد از گزينه –r براي جستجوي بازگشتي دايركتوري ها از يك دايكتوري مشخص استفاده كنيد. مانند قبل گزينه هاي مختلف مي توانند با هم تركيب شوند.

5.1.3.2: فيلتر كردن خروجي

دستور sort به طور پيش فرض خطوط خروجي را به ترتيب الفبايي مرتب مي كند.

```
thomas:~> cat people-I-like | sort

Auntie Emmy

Boyfriend
```

Dad

Grandma

Mum

My boss

ولي sort بسياري كارهاي ديگر نيز مي تواند انجام دهد. به عنوان نمونه نگاه كردن اندازه فايل. با اين دستور خروجي بر حسب اندازه فايل ها مرتب مي شود:

```
ls -la | sort -nk 5
```

نگارش قديمي sort

ممكن است نتايج مشابهي را به شكل ls -la | sort +4n بدست آوريد ولي اين يك راه قديمي است كه با استانداردهاي فعلي همخواني ندارد

.دستور sort به صورت تركيبي با برنامه uniq نيز به كار مي رود(يا sort –u) تا بعد از مرتب سازي موارد مشابه حذف شوند.

5.2: خلاصه

در اين فصل آموختيم كه چگونه مي توان چند دستور را با هم تركيب كرد و چگونه خروجي يك دستور را ميتوان به عنوان ورودي يك دستور ديگر به كار برد.

هدايت ورودي ها و خروجي ها به عنوان يك عمل معمول در ماشين هاي لينوكسي و انجام مي پذيرد. اين مكانيزم قدرتمند امكان استفاده انعطاف پذيري از اجزاي سازنده يونيكسي به ما مي دهد.

5.3:تمرين ها

اين تمرين ها نمونه هاي بيشتري از تركيب دستورات ارائه مي كند. هدف استفاده هر چه كمتر از كليد Enter مي باشد.

- دستور cut را روي خروجي يك ليست بلند از محتويات يك دايركتوري به كار ببريد تا فقط مجوزهاي فايل ها نمايش داده شود. سپس خروجي را به sort و uniq بفرستيد تا تمام خطوط تكراري حذف شوند و در نهايت از wc براي شمارش انواع مختلف مجوزهاي فايل در اين دايركتوري استفاده نماييد.
- خروجي date را در يك فايل قرار دهيد. خروجي ls را به اين فايل اضافه نماييد. اين فايل را به صندوق نامه هاي محلي خودتان ارسال كنيد (از چيزي مثل <@domain.com> استفاده نكنيد،فقط نام كاربري را وارد كنيد). اگر از bash استفاده مي كنيد يك پيام مبني بر اينكه يك نامه جديد دريافت كرده ايد مشاهده خواهيد كرد.
- دستگاه هاي در /dev را كه UID شما از آن استفاده كرده است ليست كنيد. خروجي را در less پايپ كنيد تا به طور قابل استفاده نمايش داده شوند.
- از دستور bash –x براي اجراي دايمون httpd با استفاده از اسكريپت موجود در /etc/rc.d/init.d استفاده كنيد. خطا ها را به يك فايل بفرستيد. همين كار را مجددا تكرار كنيد ولي اين بار خروجي را به يك فايل بفرستيد. تفاوت را مشاهده نماييد.

- در حال حاضر چند process اجرا کرده اید؟
- چند فايل مخفي در دايركتوري خانگي شما موجود است؟
- از locate براي يافتن مستنداتي درباره كرنل استفاده كنيد.
- فايلي كه حاوي خط زير است را پيدا كنيد:

root:x:0:0:root:/root:/bin/bash

و همچنين اين يكي:

system: root

- ببينيد با اجراي دستور زير چه اتفاقي مي افتد:

```
time; date >> time; cat < time <
```