پنج شنبه 24 مهر 1382

راهنمای کار با خط فرمان متنی لینوکس – بخش سوم

اتصال و گسترش فرامین

یکی از قابلیتهای واقعا قدرتمند پوسته فرمان، قابلیت هدایت خروجی یا ورودی یک فرمان به فرامین دیگر است. برای این منظور، همانطور که قبلا اشاره شد، از کاراکترهای ویژه استفاده میشود.

لوله بندی فرامین (Piping Commands)

کاراکتر ویژه لوله بندی کاراکتر (|) است. این کاراکتر، خروجی یک فرمان را به ورودی فرمان دیگر هدایت میکند. برای مثال :

```
$ cat /etc/passwd | sort | more
adm:x:3:4:adm:/var/adm:/sbin/nologin
Alan:x:500:500:Alan Bachumian,7852020:/home/Alan:/bin/bash
apache:x:48:48:Apache:/var/www:/sbin/nologin
bin:x:1:1:bin:/bin:/sbin/nologin
Linet:x:501:501:Linet Minasian:/home/Linet:/bin/bash
mail:x:8:12:mail:/var/spool/mail:/sbin/nologin
--More--
```

این فرمان محتویات فایل etc/passwd را خوانده و خروجی را به فرمان sort هدایت میکند. این فرمان، کاراکتر ابتدای هر سطر را گرفته و خروجی را بصورت الفبایی مرتب کرده و خروجی را به دستور more میفرستد و این دستور نیز خروجی را بصورت صفحه به صفحه نمایش میدهد.

قابلیت لوله بندی نمایش خوبی است از اینکه چگونه یونیکس، پدر لینوکس بر اساس قطعات مختلف نرم افزاری شکل گرفته است. مثلا در یونیکس ابزارهای مختلف را طوری به هم وصل میکردن که کارهای مختلفی بتوان با آنها انجام داد. مثال خوبی که در این مورد میشود زد: سالها پیش که واژه پردازهای گرافیکی و راحت مانند اکنون وجود نداشتند، کاربران باید ابتدا سند خود را بصورت متنی ایجاد کرده و سپس آنرا بوسیله ماکروهای خاصی فرمت بندی میکردند و بعد باید بررسی میکردند که چطور از آب در آمده است . برای این کار از فرمانی مانند زیر استفاده میشد:

```
$ nroff -man grep.1 | lpr
```

در دستور بالا از nroff برای فرمت کردن فایل grep.1 با استفاده از ماکروی man استفاده شده و حاصل کار با استفاده از لوله بندی به خروجی چاپگر که lpr است فرستاده شده است.

دستورات متوالی

برخی اوقات نیاز دارید که چند فرمان بصورت همزمان با استفاده از یک فرمان اجرا شوند. این کار به سادگی امکان پذیر است. کافی است پس از اتمام هر فرمان آنرا از فرمان بعدی بوسیله یک کاراکتر سم کالن (;) جدا کنید. برای مثال :

```
$ date; troff -me mytext | lpr; ls /home
```

فرامین پس زمینه

برخی دستورات برای تمام شدن نیاز به زمان دارند. برخی اوقات مایل نیستید که پوسته فرمانتان را معطل باقی بگذارید تا دستور به اتمام برسد. برای این منظور میتوانید دستور مورد نظر را با استفاده از کاراکتر آمپرسند (&) در پس زمینه اجرا کنید. برای مثال :

```
$ troff -me mytext &
```

راههایی برای مدیریت پروسه های پیش زمینه و پس زمینه وجود دارد که جلوتر درباره آنها صحبت خواهیم کرد.

توسعه فرامین

به وسیله قابلیت جانشینی فرامین میتوانید پوسته فرمان را وادار کنید تا خروجی یک فرمان را خودش تفسیر کند، بجای اینکه این کار به خود فرمان واگذار شود. در این مورد شما میتوانید خروجی استاندارد یک فرمان را بصورت آرگومان یک دستور دیگر تعیین کنید.

دو شکل قابلیت جانشینی فرامین بصورت زیر است :

$(sommand) or 'command'

برای فهمیدن این قابلیت به مثال زیر توجه کنید:

```
$ vi $(find / -print | grep xyzzy)
```

در این دستور، قبل از اجرای vi جانشینی فرامین صورت می گیرد. ابتدا دستور find از دایرکتوری ریشه شروع به کار کرده و نام تمام فایلها و دایرکتوری ها را چاپ میکند. خروجی این دستور به grep ارسال میشود و این دستور تمام آنها را که فاقد رشته xyzzy هستند را فیلتر میکند. سپس vi تمام فایلهایی را که دارای رشته xyzzy هستند را باز میکند.

توسعه عبارات حسابی

موارد زیادی وجود دارد که شما مایلید که نتایج یک جمله محاسباتی را به یک فرمان ارسال کنید . دو راه برای انجام آن وجود دارد:

$[expression] or $((expression))

برای روشن شدن مطلب به مثال زیر توجه کنید :

$ echo "Iam $[2003-1978] years old."

Iam 25 years old.

در مثال بالا، پوسته فرمان ابتدا عبارت حسابی را انجام داده و سپس نتیجه را به فرمان echo ارسال میکند.

توسعه متغییرهای محیطی

همانطور که قبلا گفتیم، متغییرهای محیطی اطلاعاتی را در مورد پوسته فرمان در بر دارند. هنگامی که یک متغییر محیطی را در یک دستور قرار میدهید، بجای اینکه نام آن چاپ شود، محتویات آن چاپ میشود :

$ ls -l $BASH

-rwxr-xr-x 1 root root 626188 Aug 24 2002 /bin/bash

در مثال بالا، دستور ls با استفاده از متغییر محیطی BASH محل آنرا چاپ میکند. در این مورد بیشتر توضیح خواهم داد.

استفاده از متغییر های محیطی

متغییرهای محیطی برای ذخیره اطلاعاتی مانند محل فایلهای پیکربندی ، صندوقهای پستی و مسیر دایرکتوری ها بکار میروند.
همچنین این متغییرها دارای مقادیری برای شکل اعلان فرمان، اندازه تاریخچه و نوع سیستم عامل نیز هستند.
برای نمایش متغییرهایی که اکنون به پوسته فرمان شما اختصاص داده شده اند، باید از دستور declare استفاده کنید.
برای نمایش محتویات هر یک ، کافی است یک علامت دلار جلوی آن قرار داده و آن را در دستورات خط فرمان استفاده کنید :

$ echo $USER

Alan

همانطور که می بینید، فرمان بالا نام کاربر فعلی سیستم را نمایش میدهد.

متغییرهای محیطی عمومی

هنگامی که یک پوسته فرمان باز میکنید، متغییرهایی وجود دارند که مقادیر آنها قبلا تخصیص داده شده است. در زیر برخی از این متغییرها نشان داده شده اند :

- BASH : محتوی مسیر کامل برنامه پوسته فرمان است. به طور معمول bin/bash .
- BASH_VERSION : شماره نسخه برنامه پوسته فرمان را نشان میدهد.
- EUID : شماره شناسایی موثر کاربر فعلی را نمایش میدهد. این مقدار هنگامی که پوسته شروع میشود، تخصیص داده میشود.
- HISTFILE : محل فایل تاریخچه فرامین را نمایش میدهد.
- HISTFILESIZE : تعداد فرامینی که تاریخچه در خود نگهداری میکند. معمولا ۱۰۰۰ است.
- HISTCMD : شماره فرمان جاری را در تاریخچه نشان میدهد.
- HOME : دایرکتوری خانگی کاربر جاری را نشان میدهد.
- HOSTTYPE : نوع معماری پردازنده کامپیوتر را نشان میدهد.
- MAIL : مسیر صندوق پستی کاربر جاری را نشان میدهد. معمولا به نام شما در /var/spool/mail قرار دارد.
- OLDPWD : مسیر قبل از دایرکتوری جاری فعلی را نشان میدهد.
- OSTYPE : نوع سیستم عامل را نشان میدهد. در مورد ما خروجی به صورت linux-gnu خواهد بود.
- PATH : لیست دایرکتوری های معرفی شده را نشان میدهد. برای اجرای یک فرمان در این دایرکتوری ها جستجو صورت میگیرد.
- PPID : شماره پروسه ای که پوسته فرمان را شروع کرده است، نمایش میدهد.
- PROMPT_COMMAND : دستوری را که هربار پیش از نمایش اعلان فرمان اجرا میشود را نشان میدهد.
- PS1 : مقدار اعلان فرمان را تخصیص میدهد. مقادیر زیادی وجود دارند که آنها را میتوانید در اعلان فرمان خود بگنجانید مانند تاریخ، زمان، نام کاربر، نام کامپیوتر و ... برخی اوقات یک فرمان به اعلان های بیشتری نیاز دارد که میتوانید از متغییرهای PS2 یا PS3 برای این کار استفاده کنید. در این مورد بیشتر توضیح خواهم داد.
- PWD : دایرکتوری جاری را نشان میدهد.
- RANDOM : با مراجعه به این متغییر یک شماره تصادفی بین ۰ و ۹۹۹۹ تولید میشود.
- SECONDS : تعداد ثانیه ای که پوسته فرمان آغاز به کار کرده است.
- UID : شماره شناسایی اصلی کاربر فعلی را نمایش میدهد. این شماره در فایل etc/passwd ذخیره شده است.

ایجاد متغییرهای محیطی خاص

از متغییرهای محیطی میتوانید برای ذخیره اطلاعاتی که معمولا در پوسته فرمان استفاده میکنید، بهره برداری کنید. شما میتوانید هر گونه متغییر محیطی به دلخواه خود ایجاد کنید. برای ایجاد موقت یک متغییر محیطی میتوانید نام متغییر و مقدار آن را جلوی اعلان فرمان تایپ کنید :

$ AB=/usr/locl/documents; export AB

مثال بالا مسیر یک دایرکتوری را به یک متغییر به نام AB اختصاص میدهد. دستور export این متغییر را به پوسته فرمان صادر میکند. بنابراین در صورتی که پوسته های فرمان دیگری نیز اجرا شوند، این متغییر در آنها موجود خواهد بود.

نکته : ممکن است توجه کرده باشید که تمام متغییرهای محیطی با حروف بزرگ تعریف شده اند. این کار یک رسم است نه یک الزام. یعنی در صورتی که نام متغییری را با حروف کوچک تعیین کنید ، باز هم کار خواهد کرد. البته توجه داشته باشید که متغییر xyz با XYZ یکی نیست.

مشکلی که در ایجاد این گونه متغییرهای محیطی وجود دارد این است که موقت بوده و با خروج از پنجره پوسته ای که این متغییر در آن تخصیص داده شده است، این متغیر پاک خواهد شد. برای اختصاص دائمی این متغییرها، باید آنها را به فایلهای پیکربندی پوسته فرمان اضافه کنید. این موضوع جلوتر توضیح داده خواهد شد.
در صورتی که مایلید متنی درست جلوی مقدار یک متغییر محیطی قرار گیرد، کافی است که متغییر را در دو پرانتز قرار داده و متن مورد نظر را جلوی آن قرار دهید . برای مثال :

$ echo ${HOME}/Documents
/home/Alan/Documents

به خاطر داشته باشید که برای استفاده از متغییرها یا باید آنها را export کنید و یا به فایل پیکربندی پوسته فرما اضافه نمایید. دستور export بسیار قابل انعطاف است. مثلا میتوانید در هنگام صادر کردن متغییر، مقدار آنرا هم تخصیص دهید :

$ export XYZ=/home/Alan/Documents

ویا میتوانید با حفظ مقادیر قبلی، مقداری را به یک متغییر اضافه نمایید :

$ export PATH=$PATH:/home/Alan/Documents

در مثال بالا، دایرکتوری home/Alan/Documents به طور موقت به متغییر PATH اضافه شده است.
در صورتی که احساس کردید دیگر به یک متغییر نیازی ندارید، میتوانید با استفاده از دستور unset آنرا پاک کنید:

$ unset XYZ

همانطور که دیدید، برای پاک کردن متغییر نیازی به علامت دلار نیست.

مدیریت پروسه های پس زمینه و پیش زمینه

در صورتی که از لینوکس در محیط شبکه ای و با استفاده از یک ترمینال متنی استفاده میکنید، پوسته فرمان تنها چیزی است که میتوانید از آن استفاده کنید و از محیطهای گرافیکی خبری نخواهد بود. در صورتی که نیاز داشته باشید در آن واحد با چندین برنامه کار کنید، این مسئله بسیار محدود کننده خواهد بود.
با اینکه پوسته فرمان محیطی گرافیکی برای اجرای برنامه ها ندارد، ولی قابلیتی دارد که با استفاده از آن میتوانید برنامه های فعال را بین پس زمینه و پیش زمینه جابجا نمایید. با این وسیله میتوانید تعداد زیادی برنامه را در یک زمان درحال اجرا داشته باشید و بین آنها حرکت کنید.
راههای گوناگونی برای قرار دادن یک برنامه در پس زمینه وجود دارد. قبلا اشاره کردیم که با اجرای برنامه ای که به آخر آن یک کاراکتر آمپرسند (&) اضافه شده است، در پس زمینه قرار میگیرد. روش دیگر استفاده از دستور at برای اجرای برنامه ها بصورتی که به پوسته متصل نباشند ، است.
برای توقف اجرای یک فرمان و قرار دادن آن در پس زمینه، از کلیدهای Ctrl+z استفاده کنید. پس از اینکه اجرای دستور متوقف شد، با استفاده از دستور fg میتوانید آنرا به پیش زمینه آورده، استفاده کنید و یا با دستور bg آنرا در پس زمینه بکار بگیرید.

شروع پروسه های پس زمینه

در صورتی که برنامه هایی دارید که مایلید در هنگام کار کردن شما در پس زمینه اجرا شوند، پس از دستور ، یک علامت آمپرسند (&) در پایان آن اضافه کنید. برای مثال :

$ find /usr -print > /home/Alan/usrfiles &

این دستور تمام فایلهای موجود در دایرکتوری usr لینوکس شما را در فایلی به نام usrfiles ذخیره میکند. علامت آمپرساند باعث میشود که این فرمان در پس زمینه اجرا شود. برای دیدن اینکه چه برنامه هایی در پس زمینه در حال اجرا هستند، از دستور jobs استفاده کنید:

$ jobs
[1]- Stopped mc
[2]+ Stopped vi
[3] Running find /usr -print >usrfiles &

همانطور که در خروجی فرمان بالا مشاهده میکنید، سه برنامه vi ، mc و دستور find در حال اجرا در پس زمینه هستند. علامت مثبت در کنار برنامه دوم نشان میدهد که این آخرین پروسه ای است که درحالت پس زمینه اجرا شده است و علامت منفی نشاندهنده پروسه ای است که قبل از آخرین پروسه، در پس زمینه قرار داده شده است. بعلت اینکه برنامه های اول و دوم برای کارکرد به خروجی ترمینال نیاز دارند تا زمانی که در حالت پیش زمینه اجرا شوند، متوقف باقی خواهند ماند. ولی برنامه find که به خروجی ترمینال نیازی ندارد، در حال اجرا میباشد.

نکته : برای نمایش شماره پروسه برنامه های پس زمینه ، میتوانید گزینه l را به فرمان jobs اضافه نمایید. در صورتی که از دستور ps برای نمایش پروسه های فعال استفاده کنید، میتوانید ببینید که کدامیک از آنها دستوری است که در پس زمینه در حال اجراست.

نوشته شده توسط کاوشگر در ساعت 11:14pm
نظرات [0]

آخرین اخبار فناوري اطلاعات برگرفته از سایت ایتنا www.itna.ir