



آزمایشگاه سیستم عامل

دستورکار ۳: کار با ترمینال لینوکس (۱)

- دستور **cat** برای ایجاد یا نمایش محتویات فایل‌ها استفاده می‌شود.
`cat file_name`
- دستور **head** محتویات ابتدای فایل را نمایش می‌دهد.

لیست برخی از سویچ‌های دستور head	
-n	تعداد خطوط که باید از اول فایل نمایش داده شوند را مشخص می‌کند.

مثال:

```
head -n 7 filename.txt
```

- دستور **tail** محتویات انتهای فایل را نمایش می‌دهد.

لیست برخی از سویچ‌های دستور tail	
-n	تعداد خطوط که باید از انتهای فایل نمایش داده شوند را مشخص می‌کند.

- دستور **wc** تعداد سطرها، کلمات و حروف یک فایل متنی را به ترتیب از چپ به راست نمایش می‌دهد.

`wc [option] {file-name}`

لیست برخی از سویچ‌های دستور wc	
-l	تعداد سطرها را نمایش می‌دهد.
-w	تعداد کلمات را نمایش می‌دهد.
-c	تعداد حروف را نمایش می‌دهد.

مثال:

```
wc -l a.txt  
wc -w a.txt  
wc -c a.txt
```

- دستور **cut** بخش‌هایی از هر سطر را از درون یک متن و یا یک فایل متنی بیرون می‌کشد.

لیست برخی از سویچ‌های دستور cut	
-b	تعداد بایت‌ها را مشخص می‌کند.
-c	تعداد حروف را مشخص می‌کند.
-d	جداکننده (delimiter) را مشخص می‌کند. جداکننده پیش‌فرض tab است.
-f	برای مشخص کردن فیلد مورد نظر به کار می‌رود.

مثال:

```
cut -b 1 a.txt
cut -d ' ' -f 2 a.txt
```

اولین بایت از هر سطر فایل را بیرون می‌کشد.

- دستور **less** اطلاعات فایل را صفحه به صفحه نمایش می‌دهد و می‌توانیم خروجی را در دو جهت پیمایش کنیم. همچنین امکان جستجو برای عبارتی خاص را به ما می‌دهد. برای این منظور کفایست ابتدا کاراکتر / را وارد نموده و به دنبال آن عبارت مورد جستجو را تایپ می‌کنیم.

`less {file_name}`

- دستور **more** هم مانند less اطلاعات فایل را صفحه به صفحه نمایش می‌دهد اما more امکان پیمایش تنها در یک جهت را نشان می‌دهد.

✓ با فشردن دکمه q می‌توانیم از محیط آن خارج شویم.

- دستور **grep** یک رشته را جستجو می‌کند. این جستجو می‌تواند در محتویات یک فایل و یا در خروجی یک دستور انجام شود.

`grep {pattern} [file_name]`

لیست برخی از سویچ‌های دستور grep	
-c	تعداد خطوط پیدا شده را نمایش می‌دهد.
-r	تمامی فایل‌های داخل یک دایرکتوری را به صورت بازگشتی جستجو می‌نماید.
-i	جستجو به صورت case sensitive انجام نمی‌شود.

- دستور **echo** برای نمایش دادن پیام بر روی خروجی استاندارد بکار می‌رود. در آخر عبارت \n می‌گذارد یعنی سر خط بعد می‌رود.

لیست برخی از سویچ‌های دستور echo	
-n	آخر رشته \n نمی‌گذارد.
-e	تفسیر backslash escape را فعال می‌کند.
-E	تفسیر backslash escape را غیرفعال می‌کند.

- تغییر مسیر خروجی استاندارد:

`{command}>{file_name}`

در این حالت خروجی دستور به جای نمایش روی خروجی استاندارد (صفحه مانیتور) در فایل مورد نظر ذخیره می‌شود.

✓ چنانچه فایلی که تعیین می‌کنیم تا خروجی دستور در آن ذخیره شود، در سیستم وجود نداشته باشد ابتدا فایل در دایرکتوری جاری ایجاد می‌شود. سپس خروجی دستور در آن ذخیره می‌گردد. اما، در صورتی که فایلی با آن نام از قبل در سیستم وجود داشته باشد، خروجی حاصل از دستور جایگزین محتویات قبلی آن فایل می‌شود.

- چنانچه از `>>` استفاده نماییم خروجی حاصل از اجرای دستور در انتهای فایل اضافه می‌شود (`append`) و اطلاعات قبلی فایل از بین نمی‌رود.

مثال:

```
echo "this is a test file" > test.txt
ls
cat test.txt
echo "this is a new line" > test.txt
ls
cat test.txt
echo "this is a test file" >> test.txt
cat test.txt
```

- چنانچه بخواهیم خروجی حاصل از اجرای چند دستور را در فایل ذخیره نماییم به شکل زیر عمل می‌کنیم:

`(command1 ; command2 ; ... ; commandn;) > {file_name}`

مثال:

```
(ls -l ; echo "Hello" ; echo "test" ;) > test.txt
cat test.txt
```

- چنانچه بخواهیم یک فایل ایجاد کنیم و محتویات مورد نظر را در آن وارد نماییم به شکل زیر عمل می‌کنیم:

`cat > {file_name}`

پس از اجرای این دستور ابتدا فایل ایجاد می‌شود سپس مکان‌نما به خط بعدی رفته و منتظر دریافت ورودی از کاربر می‌ماند. پس از وارد نمودن اطلاعات مورد نظر با فشردن میانبر `Ctrl+D` فایل را ذخیره می‌نماییم.

- خواندن ورودی استاندارد از یک فایل:

`{command}<{file_name}`

در این حالت `shell` اطلاعات ورودی‌اش را به جای صفحه کلید از یک فایل می‌خواند.

- تغییر مسیر ورودی و خروجی استاندارد:

`{command}<{source}>{destination}`

در این حالت ورودی دستور را از source می‌خواند و خروجی آن را در destination ذخیره می‌کند.

- **Pipe file**: خروجی استاندارد را به ورودی استاندارد می‌برد. هنگامی که خروجی یک دستور را به عنوان ورودی به دستور دیگر بدهیم از pipe file استفاده می‌نماییم علامت آن | است.

{command1} | {command2}

مثال:

```
ls /etc/ | less
```

تمرین

۱- یک فایل به نام grade.txt در دایرکتوری خانگی‌تان بسازید به طوری که دارای چهار ستون به فرمت زیر باشد. (جداکننده ستون‌ها tab است.)

نمره	نام	نام خانوادگی	شماره دانشجویی
------	-----	--------------	----------------

۲- یک فایل به نام test.txt در دایرکتوری خانگی‌تان بسازید که حاوی اسامی روزهای هفته باشد. سپس محتویات فایل passwd که در دایرکتوری /etc واقع است را به آن اضافه کنید.

۳- در قالب یک دستور هر فایلی در مسیر /etc که در نامش عبارت init دارد را لیست کنید.

۴- در قالب یک دستور فیلدهای موجود ۱۰ خط اول فایل /etc/passwd را از هم جدا کرده و در صفحه ترمینال چاپ کنید.

۵- تعداد حروف ستون اول ۱۰ خط انتهایی فایل /etc/group را محاسبه کنید.