

سوال ۱:

در حقیقت فایل سیستم شامل میلیون ها [inode](#) می باشد که با ایجاد هر فایل در لینوکس این فایل به یکی از این شماره ها ( inode) لینک می شود. در صورتی که بخواهیم یک فایل با دو نام متفاوت داشته باشیم به گونه ای که فایل مورد نظر از دو مسیر قابل دسترسی باشد از مفهومی بنام hard link استفاده می شود.

مثال: در فولدری بنام folder1 اقدام به ایجاد یک فایل بنام originalfile می نماییم. با دستور nano در این فایل عبارت هایی را وارد و ذخیره می کنیم.

```
Mkdir folder1
```

```
Cd folder 1
```

```
Touch originalfile
```

```
Nano originalfile
```

```
/cd
```

```
Mkdir folder2
```

```
Cd folder2
```

```
Ln /folder1/originalfile /folder2/linkfile
```

با این کدها یک لینک از نوع Hard ایجاد شد.. یعنی این دو فایل در حقیقت یک فایل هستند اما با دو نام متفاوت. هر تغییری که در فایل originalfile انجام شود این تغییر در فایل linkfile نیز اعمال خواهد شد. بالعکس. نتیجه گیری نهایی اینکه هر تغییری که در یکی از فایل ها انجام شود آن تغییر در فایل دوم نیز اعمال خواهد شد.

سمبلیک لینک مانند این است که شورتکاتی ایجاد شده است که به فایل اصلی اشاره می کند، اگر فایل اصلی حذف شود لینک دیگر به درستی کار نمی کند، در صورتی که اگر هارد لینک بگیریم، فایل با پاک کردن فایل اصلی خراب نمیشود و فرق عمده ی دیگری که بین هارد لینک و سمبلیک لینک وجود دارد این است که از دایرکتوری نمی توان هارد لینک گرفت ولی سافت لینک امکانش وجود دارد.

سوال ۲:

لینک های سمبولیک وابسته به مکان فایل اصلی ساخته می شود،در واقع اگر از یک فایل سافت لینک و سافت لینک وابسته بگیریم با جابجایی فایل اصلی،سافت لینک وابسته مشکل دار خواهد شد

سوال ۳:

ln-s:باعث ایجاد لینک های سخت

ln-p:امکان ایجاد پیوند سخت در پیوند های سمبولیک

ln-t:مشخص کردن محل دایرکتوری که می خواهیم در آن لینک بسازیم

ln-f:فایلی که در مقصد است را پاک می کند

سوال ۴:

What is chmod:اجازه تغییر دسترسی ها را به ما می دهد،در واقع اجازه دسترسی فایل و پوشه ها را تغییر میدهد.