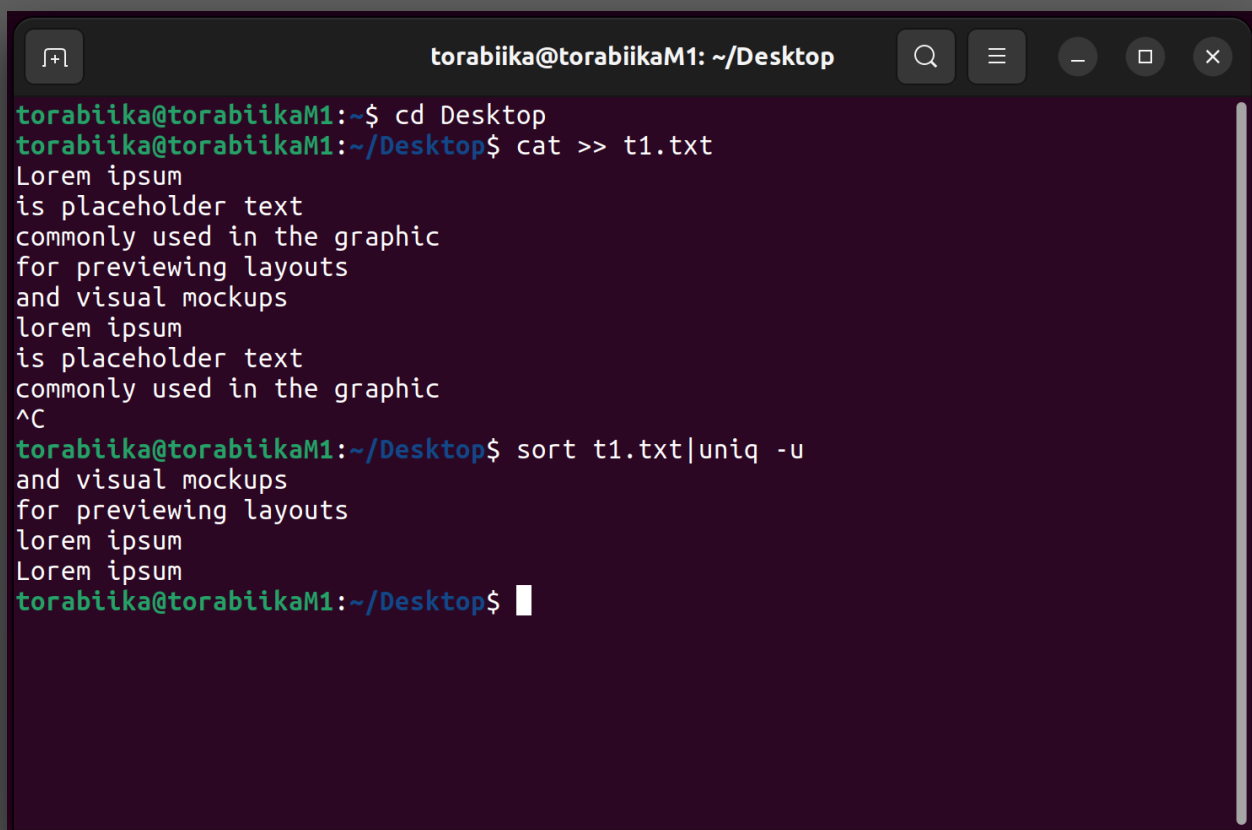


## تمرین دوم درس آزمایشگاه سیستم عامل – گروه ۹

---

تمرین اول -

- دستور **sort** در لینوکس، کار مرتب‌سازی رکوردهای موجود در یک فایل را انجام می‌دهد و بدون اینکه تغییرات را در خود فایل نهایی کند، خروجی را در ترمینال به ما نمایش می‌دهد.  
این دستور، کار بررسی و مرتب‌سازی محتوا را خط به خط انجام می‌دهد و با اینکه در حالت پیش‌فرض، براساس **ASCII** کار می‌کند اما می‌توان با استفاده از ساختار **sort -[Option]**، با ترتیب‌های دیگری، مثل ترتیب عددی و... نیز کار کرد.
  - دستور **uniq** در یک فایل، به طور کلی به دنبال خطوط تکراری می‌گردد و با استفاده از ساختار **uniq -[Option]**، خروجی‌های متفاوتی بر این اساس در ترمینال نمایش می‌دهد.
  - دستور **sort t1.txt | uniq -u** که در صورت تمرین ذکر شده، با دستور اول، خطوط فایل **t1.txt** را مرتب کرده و با دریافت آن بعنوان ورودی دستور بعدی، خطوط تکراری را حذف و خروجی را در ترمینال نمایش می‌دهد.
- در اسکرین‌شات زیر، ابتدا به **Desktop** رفته، یک فایل متنی با نام **t1** ایجاد می‌کنیم و خطوط محتوا را در آن قرار می‌دهیم، سپس با استفاده از دستور موردنظر، مشاهده می‌کنیم که خروجی براساس حروف الفبا و بصورت **Case Sensitive** و با حذف خطوط تکراری، در خروجی ترمینال ما نمایش داده شده است.



```
torabiika@torabiikaM1: ~/Desktop
torabiika@torabiikaM1:~$ cd Desktop
torabiika@torabiikaM1:~/Desktop$ cat >> t1.txt
Lorem ipsum
is placeholder text
commonly used in the graphic
for previewing layouts
and visual mockups
Lorem ipsum
is placeholder text
commonly used in the graphic
^C
torabiika@torabiikaM1:~/Desktop$ sort t1.txt|uniq -u
and visual mockups
for previewing layouts
Lorem ipsum
Lorem ipsum
torabiika@torabiikaM1:~/Desktop$
```

! نکته: دستور **uniq**، به تنهایی، وقتی اقدام به حذف خطوط تکراری می‌کند که این خطوط پشت سر هم قرار گرفته باشند، بنابراین بهتر است که مثل بالا، بعد از دستور **sort** از آن استفاده شود تا این دستور ابتدا خطوط را پشت سر هم قرار داده و خروجی آن بعنوان ورودی **uniq** استفاده شود.

همچنین دستور **sort -u** جایگزین کوتاه و مناسبی برای دستور بالاست.

تمرین دوم -

- دستور **grep** در لینوکس برای جستجوی یک **String** یا **Regular Expression** در یک فایل متنی به کار می‌رود. همانطور که سوال از ما خواسته، می‌توانیم با تعریف یک رشته و استفاده از این دستور، در فایل‌های **html** به دنبال تگ‌های باز و بسته **<head>** بگردیم.
- دستور **cp** در لینوکس برای کپی فایل‌ها و دایرکتوری‌ها استفاده می‌شود. در این سوال بعد از پیدا کردن فایل‌های موردنظر خود به وسیله دستور **grep**، از این دستور برای کپی کردن آن در **Desktop** استفاده می‌کنیم.
- برای حل این سوال، از عملیات **Pipe** تو در تو استفاده می‌کنیم که با عملگر **|** انجام می‌شود. ابتدا یک فیلتر برای پیدا کردن تگ **<head>** اعمال کرده، سپس فایل‌هایی با آن تگ را در یک دایرکتوری جدید قرار می‌دهیم و در بین آن‌ها فیلتر بعدی که قرار است به دنبال **</head>** بگردد را اعمال می‌کنیم و در نهایت فایل‌های انتخاب شده را در آدرس نهایی کپی می‌کنیم.

```
torabiika@torabiikaM1: ~/Desktop/OSLab02
torabiika@torabiikaM1:~/Desktop$ ls
t1.txt
torabiika@torabiikaM1:~/Desktop$ mkdir OSLab02
torabiika@torabiikaM1:~/Desktop$ cd OSLab02
torabiika@torabiikaM1:~/Desktop/OSLab02$ mkdir dir01
torabiika@torabiikaM1:~/Desktop/OSLab02$ cat >> html01.html
<head>
</head>
<body>
</body>
^C
torabiika@torabiikaM1:~/Desktop/OSLab02$ cat >> html02.html
<head> </head>
^C
torabiika@torabiikaM1:~/Desktop/OSLab02$ cat >> html03.html
<head>
<meta charset="UTF-8">
</head>
<body> </body>
^C
torabiika@torabiikaM1:~/Desktop/OSLab02$ cat >> html04.html
<head>
<body>
</body>
^C
torabiika@torabiikaM1:~/Desktop/OSLab02$ cat >> html05.html
</head>
<body>
^C
torabiika@torabiikaM1:~/Desktop/OSLab02$ cat >> html06.html
<body>
</body>
^C
torabiika@torabiikaM1:~/Desktop/OSLab02$ ls
dir01      html02.html  html04.html  html06.html
html01.html  html03.html  html05.html
torabiika@torabiikaM1:~/Desktop/OSLab02$
```

در ابتدا یک دایرکتوری با نام **OSLab02** ساخته سپس درون آن، دایرکتوری **dir01** را ساخته و همچنین چند فایل **html** با ویژگی‌های مختلف تعریف می‌کنیم، سپس دستور موردنظر را اجرا می‌کنیم تا فایل‌ها با ویژگی‌های موردنظر در **Desktop** کپی شوند.

طبق فایل‌هایی که ساختیم، بعد از اجرای دستور، انتظار داریم که سه فایل اول انتخاب شوند.

دستور برای عملیات خواسته شده به شرح زیر است:

```
grep -lr "<head>" html0*.html | xargs cp -t '/home/torabiika/Desktop/OSLab02/dir01' | grep -lr "</head>" '/home/torabiika/Desktop/OSLab02/dir01' | xargs cp -t '/home/torabiika/Desktop'
```

۱. **Pipe اول:** فایل هایی که تگ <head> دارند را پیدا می کند.

**grep -r**، برای چک کردن فایل ها به صورت بازگشتی است. (توضیحات بیشتر: **man grep**)

**grep -l**، برای کنترل خروجی دستور است و با اولین **match** شدن، اسکن فایل ورودی را متوقف می کند. (توضیحات بیشتر: **man grep**)

همچنین چون در مسیر خود فولدر قرار داریم، صرفاً اسم فایل ها را به شکل **wildcard** وارد می کنیم.

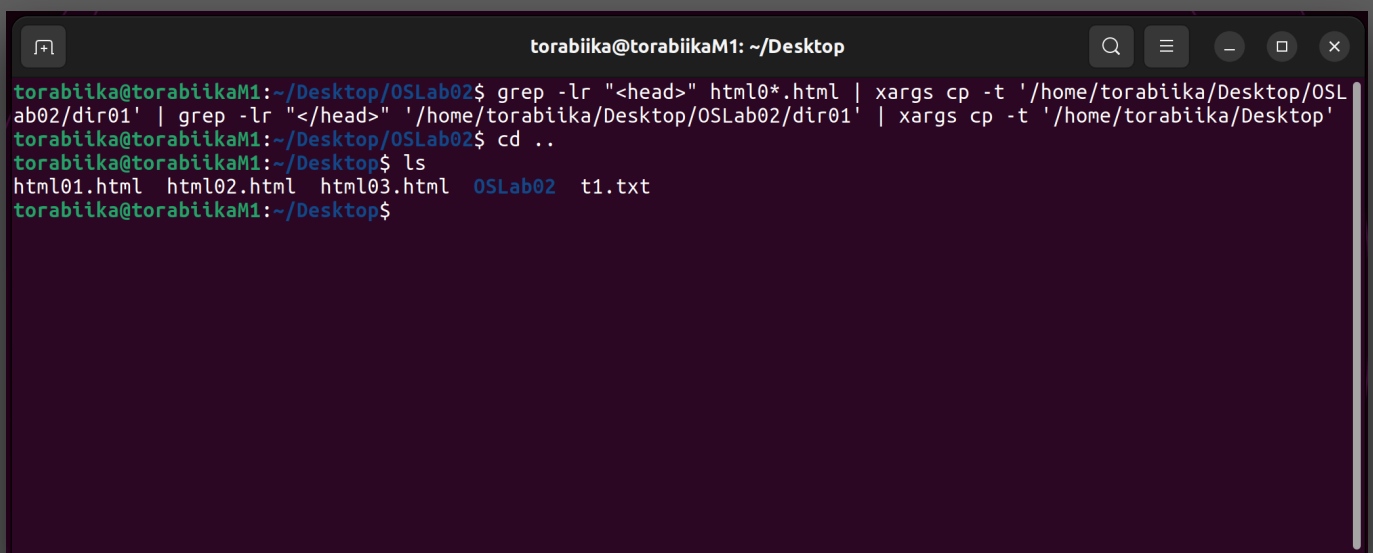
۲. **Pipe دوم:** فایل های منتخب را در دایرکتوری **dir01** کپی می کند.

دستور **xargs** در اینجا، خروجی دستور اول را بعنوان یک آرگومان **Standard Input** به دستور بعدی می فرستد.

**cp -t**، برای کپی کردن همه آرگومان های **SOURCE** در دایرکتوری موردنظر است. (توضیحات بیشتر: **man cp**)

۳. **Pipe سوم:** فایل هایی که تگ </head> دارند را از دایرکتوری **dir01** پیدا می کند.

۴. **Pipe چهارم:** آدرس دایرکتوری **Desktop** بعنوان مقصد فایل ها تعریف می شود و فایل های نهایی شده در آن مسیر کپی می شوند.



```
torabiika@torabiikaM1: ~/Desktop
torabiika@torabiikaM1:~/Desktop/OSLab02$ grep -lr "<head>" html0*.html | xargs cp -t '/home/torabiika/Desktop/OSLab02/dir01' | grep -lr "</head>" '/home/torabiika/Desktop/OSLab02/dir01' | xargs cp -t '/home/torabiika/Desktop'
torabiika@torabiikaM1:~/Desktop/OSLab02$ cd ..
torabiika@torabiikaM1:~/Desktop$ ls
html01.html  html02.html  html03.html  OSLab02  t1.txt
torabiika@torabiikaM1:~/Desktop$
```

همانطور که مشاهده می شود، سه فایل اولی که در **OSLab02** تعریف کرده بودیم و واجد شرایط صورت سوال بودند، با موفقیت در دایرکتوری **Desktop** کپی شدند.