

دانشگاه گیلان

گزارش کار تمرین دوم آزمایشگاه سیستم های عامل

استاد: آرش یوسفی

گروه ۶

سوال ۱:

فایل سیستم در هر خانواده از سیستم عامل‌ها به طور معمول متفاوت است.

ویندوز از FAT و NTFS به عنوان فایل سیستم‌های خود استفاده می‌کند در حالی که لینوکس فایل سیستم‌های مختلفی دارد.

برخلاف ویندوز، لینوکس از طریق درایو تحت شبکه قابل بوت شدن است، همچنین همه چیز در لینوکس یا پردازش است یا فایل.

لینوکس دو نوع اصلی پارتیشن دارد که یکی پارتیشن داده (data partition) و دیگری پارتیشن تعویض (swap partition) است. به موجب وجود swap partition هرگز حافظه شما در لینوکس پر نمی‌شود.

برای ابزارهای ریکاوری و بازگردانی فایل، روی فایل سیستم ویندوز ابزارهای بسیار محدودی وجود دارند که قابل استفاده‌اند؛ اما ابزارهای ریکاوری و بازگردانی بسیاری بر پایه UNIX وجود دارند که روی فایل سیستم‌های لینوکس قابل استفاده‌اند. هر دو سیستم عامل ویندوز و لینوکس فایل‌ها را بر اساس دیسک در هرمی از دایرکتوری‌ها قرار می‌دهند. این دایرکتوری‌ها معمولاً در GUI فولدر یا پوشه نامیده می‌شوند و کل هرم نیز فایل سیستم نامیده می‌شود. معماری ویندوز و لینوکس در برخی جاها یکسان است، اما وقتی درباره ابزارها صحبت می‌شود تفاوت چشم‌گیری باهم دارند.

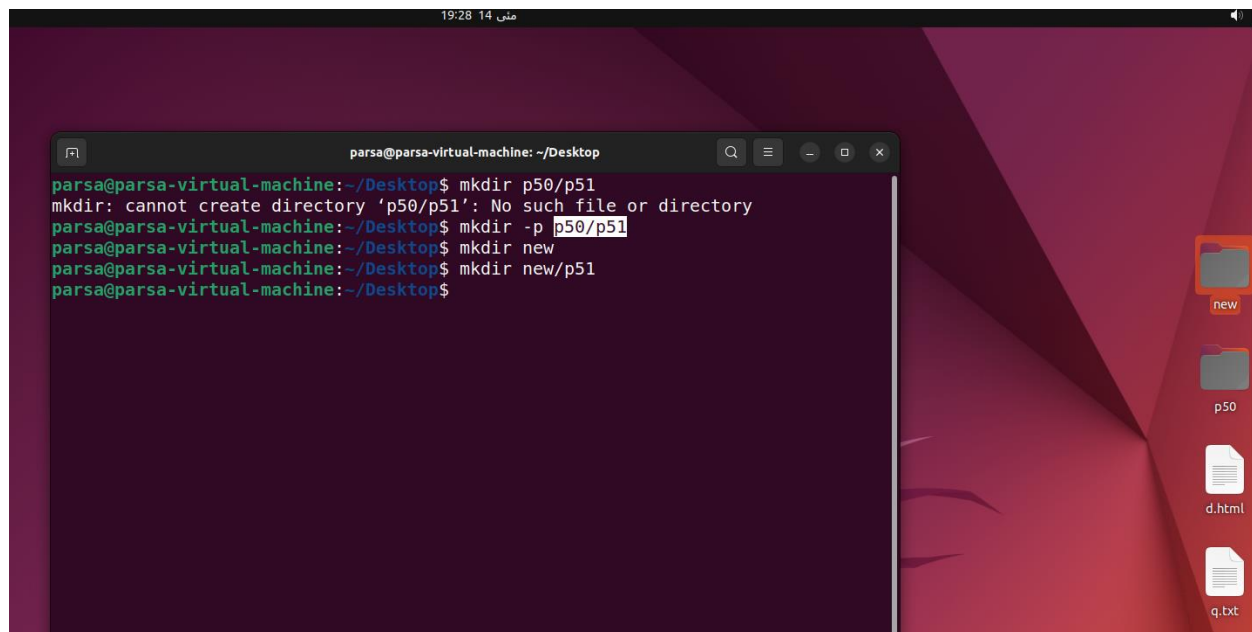
در سخت‌افزارهای IBM یا Intel هم ویندوز و هم لینوکس از Master Block Record و Master Boot Record MBR استفاده می‌کنند. این قسمتی از دیسک است که وظیفه بوت کردن سیستم عامل در آن ذخیره می‌شود. همچنین برای تعیین وضعیت پارتیشن‌بندی دیسک مورد استفاده قرار می‌گیرد. در هر دو مورد ویندوز و لینوکس در بیشترین حالت یک فایل سیستم در هر پارتیشن وجود دارد.

هر فایل سیستمی در ویندوز یک حرف نشان دهنده درایو دارد؛ مانند: C. در لینوکس هر فایل سیستم دارای یک دیوایس است، مانند: dev,hda1 که یعنی Hard Disk A part 1 که به صورت یک فایل نمایش داده می‌شود. چنین فایلی یک فایل دیوایسی است از آنجا که dev بخشی از نام آدرس است. همچنین فایل دیوایس یک فایل text عادی نیست، بلکه یک فایل ویژه است. از آنجا که دیسک‌ها دیوایس‌های بلاک هستند اینچنین فایلی به صورت "فایل دیوایس ویژه بلاک" نامیده می‌شود.

سوال ۲:

فرق بین این دو دستور در این مورد می‌باشد که با استفاده از `mkdir p50/p51` فقط در صورتی می‌توانیم دایرکتوری `p51` را ایجاد کنیم که از قبل دایرکتوری `p50` موجود بوده باشد. اما با استفاده از دستور `mkdir -p p50/p51` اگر دایرکتوری `p50` نیز وجود نداشته باشد، اول آن را ایجاد، بعد دایرکتوری `p51` را درون آن ایجاد می‌کند.

قابل توجه است که در صورتی که دایرکتوری `p50` در محیط وجود نداشته باشد، دستور اول به `error` ختم می‌شود. اما اگر وجود داشته باشد خطایی رخ نمی‌دهد.

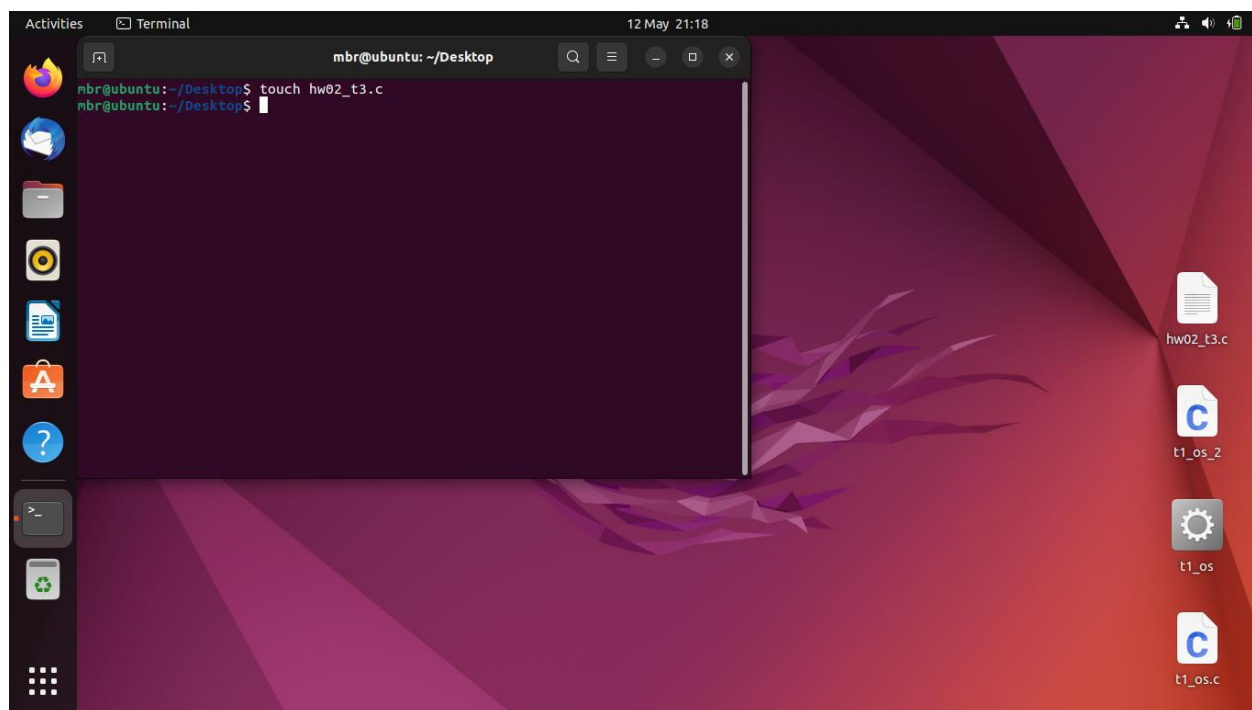


```
parsa@parsa-virtual-machine: ~/Desktop
parsa@parsa-virtual-machine:~/Desktop$ mkdir p50/p51
mkdir: cannot create directory 'p50/p51': No such file or directory
parsa@parsa-virtual-machine:~/Desktop$ mkdir -p p50/p51
parsa@parsa-virtual-machine:~/Desktop$ mkdir new
parsa@parsa-virtual-machine:~/Desktop$ mkdir new/p51
parsa@parsa-virtual-machine:~/Desktop$
```

سوال ۳:

در تمرین پیشین، پیشنیاز های کامپایل زبان در لینوکس را اضافه کرده و مراحل مورد نیاز را انجام دادیم. در این تمرین طبق مراحل زیر و تمرین خواسته شده، مطابق با گزارش زیر است:

طبق تصویر صفحه بعد، ابتدا با دستور TOUCH فایل به نام hw02_t3.c (پسوند c به منظور ایجاد فایل به زبان C ایجاد است، با این وجود می دانیم که با توجه به تمرین پیشین پسوند تاثیری در شناخت سیستم عامل لینوکس بر فایل ایجاد شده ندارد) ایجاد می کنیم.



سپس با توجه به تمرین خواسته شده، قطعه کدی باید بنویسیم که در مرحله اول از ورودی مقدار n را گرفته، سپس در n خط string (رشته) بگیرد و در پایان concatenation (به صورت یک رشته پیوسته) نتیجه را در ترمینال چاپ کند.

سورس کد مربوطه به همراه کامنت گذاری به صورت عکس در زیر ارائه شده است.

```

1  // OSLab MBR GU
2  #include <stdio.h>
3  #include <string.h>
4
5  #define MAX_LENGTH 1000
6
7  int main()
8  {
9      int n, i;
10     // Array lists for get strings from input and result for concatenation of strings
11     char str[MAX_LENGTH], result[MAX_LENGTH];
12
13     // If you want to see guide in terminal
14     // printf("Enter the number of strings: ");
15     // get n from input
16     scanf("%d", &n);
17     // in the beginning for the first value of result array we set \0
18     result[0] = '\0';
19     // get string input for n time
20     for (i = 0; i < n; i++)
21     {
22         // If you want to see guide in terminal
23         // printf("Enter string %d: ", i + 1);
24         // get string from input and store to the str
25         scanf("%s", str);
26
27         // For make concatenation of all strings together
28         strcat(result, str);
29     }
30
31     // show Result at the end on terminal with comment
32     // printf("Result: %s\n", result);
33
34     // show just Result at the end on terminal
35     printf("%s\n", result);
36     return 0;
37 }
38

```

سپس مطابق با تمرین پیشین فایل اجرایی را با استفاده از دستور زیر می‌سازیم:

```
$ gcc hw02_t3.c -o hw02_t3
```

حال با استفاده از دستور زیر، فایل اجرایی ایجاد شده را اجرا می‌کنیم:

```
./hw02_t3
```

با اجرای برنامه، در خط اول مقدار n که در این مثال مقدار ۳ وارد شده است را وارد می‌کنیم؛ و در خط‌های بعدی (مطابق با ورودی n یعنی ۳) به ترتیب رشته‌های `we`، `lc`، `ome` را وارد می‌کنیم که نتیجه نهایی به صورت `welcome` نمایش داده می‌شود.

دستورات:

3

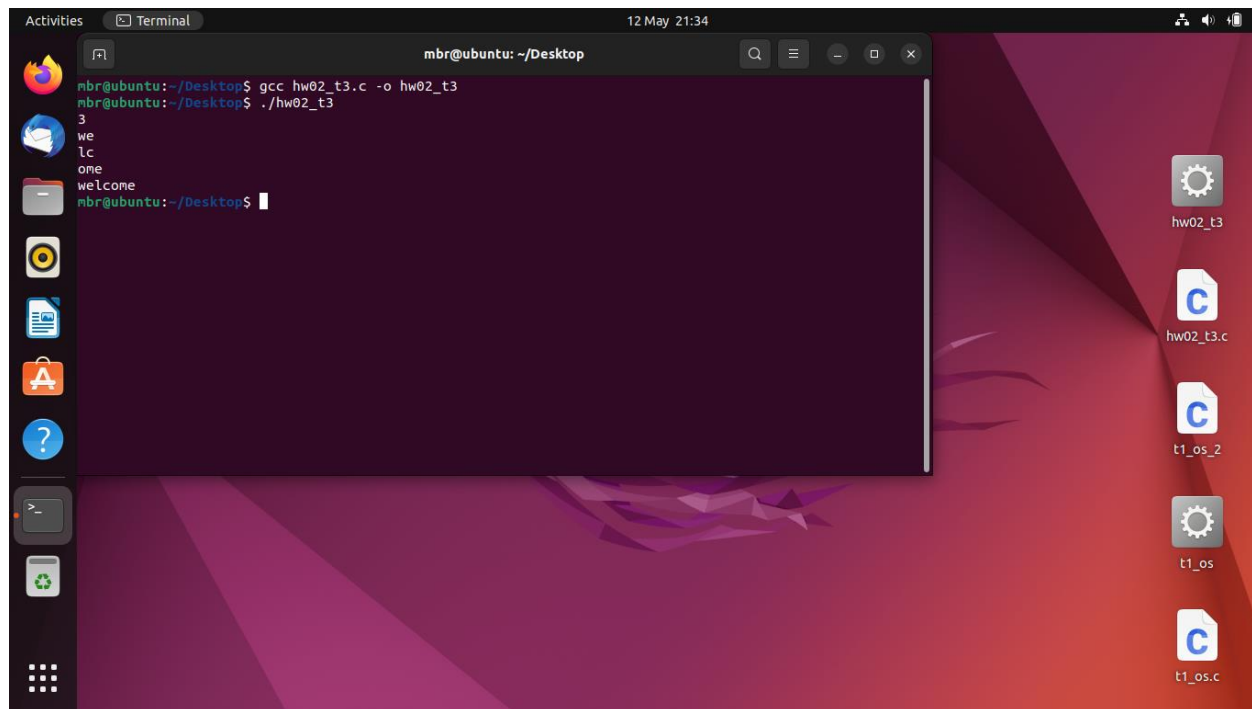
we

lc

ome

نتیجه:

welcome



The screenshot shows a Linux desktop with a purple and red background. A terminal window is open, displaying the following commands and output:

```
mbr@ubuntu: ~/Desktop
mbr@ubuntu:~/Desktop$ gcc hw02_t3.c -o hw02_t3
mbr@ubuntu:~/Desktop$ ./hw02_t3
3
we
lc
ome
welcome
mbr@ubuntu:~/Desktop$
```

The desktop has several icons on the right side: `hw02_t3`, `hw02_t3.c`, `t1_os_2`, `t1_os`, and `t1_os.c`. The terminal window title is `mbr@ubuntu: ~/Desktop`.