



به نام خدا



## دستور کار کارگاه مبانی کامپیوتر و برنامه‌نویسی

### جلسه نهم

#### مباحث ویژه

در این جلسه قصد داریم مروری برای مباحثی ویژه خارج از آنچه در کلاس آموخته‌اید داشته باشیم. در اولین قسمت قصد داریم از gcc برای کامپایل کردن برنامه به صورت دستی استفاده کنیم. همانطور که در درس آموخته‌اید عملیات کامپایل کردن از مراحل زیر تشکیل شده است:

۱. پیش پردازش

۲. کامپایل کردن

۳. اسمبلر

۴. لینکر

این عملیات‌ها غالباً به صورت یکجا توسط یک برنامه (کامپایلر) صورت می‌گیرند. یکی از کامپایلرهای موجود gcc می‌باشد که ما از آن استفاده خواهیم کرد. برای شروع با دستور زیر وارد command prompt شوید:

```
crtl + R  
cmd
```

در محیط command prompt شما می‌توانید تمام آنچه که در محیط گرافیکی انجام می‌دهید را انجام دهید. یکی از ویژگی‌های این محیط قابلیت‌های آن می‌باشد که از محیط گرافیکی بسیار بیشتر است. در ابتدا اندکی با این محیط بیشتر آشنا می‌شویم. به مانند محیط گرافیکی، command prompt هم در یک پوشه از سیستم شما اجرا می‌شود.

پوشه‌ای که در آن قرار دارید را می‌توانید در خط فرمان ببینید:

```
C:\Users\parham>
```

با دستور زیر می‌توانید محتوای این پوشه را ببینید:

```
dir
```

```
Volume in drive C has no label.  
Volume Serial Number is 1466-24A8
```

```
Directory of C:\Users\parham
```

```
11/28/2019  06:31 AM    <DIR>          .
11/28/2019  06:31 AM    <DIR>          ..
11/28/2019  05:24 PM                9,843 .bash_history
10/03/2019  07:08 PM                62 .gitconfig
10/03/2019  05:07 PM    <DIR>          .ssh
11/03/2019  09:04 AM            11,133 .viminfo
11/15/2019  08:39 PM    <DIR>          3D Objects
11/15/2019  08:39 PM    <DIR>          Contacts
11/24/2019  12:46 AM    <DIR>          Desktop
11/28/2019  05:27 PM    <DIR>          Documents
11/28/2019  05:29 PM    <DIR>          Downloads
07/15/2019  12:14 PM    <DIR>          Dropbox
11/15/2019  08:39 PM    <DIR>          Favorites
11/15/2019  08:39 PM    <DIR>          Links
11/15/2019  08:39 PM    <DIR>          Music
11/15/2019  08:40 PM    <DIR>          OneDrive
11/15/2019  08:39 PM    <DIR>          Pictures
11/15/2019  08:39 PM    <DIR>          Saved Games
11/15/2019  08:39 PM    <DIR>          Searches
11/15/2019  08:39 PM    <DIR>          Videos
11/28/2019  06:33 AM    <DIR>          wekafiles

        3 File(s)                21,038 bytes
       18 Dir(s)  13,159,411,712 bytes free
```

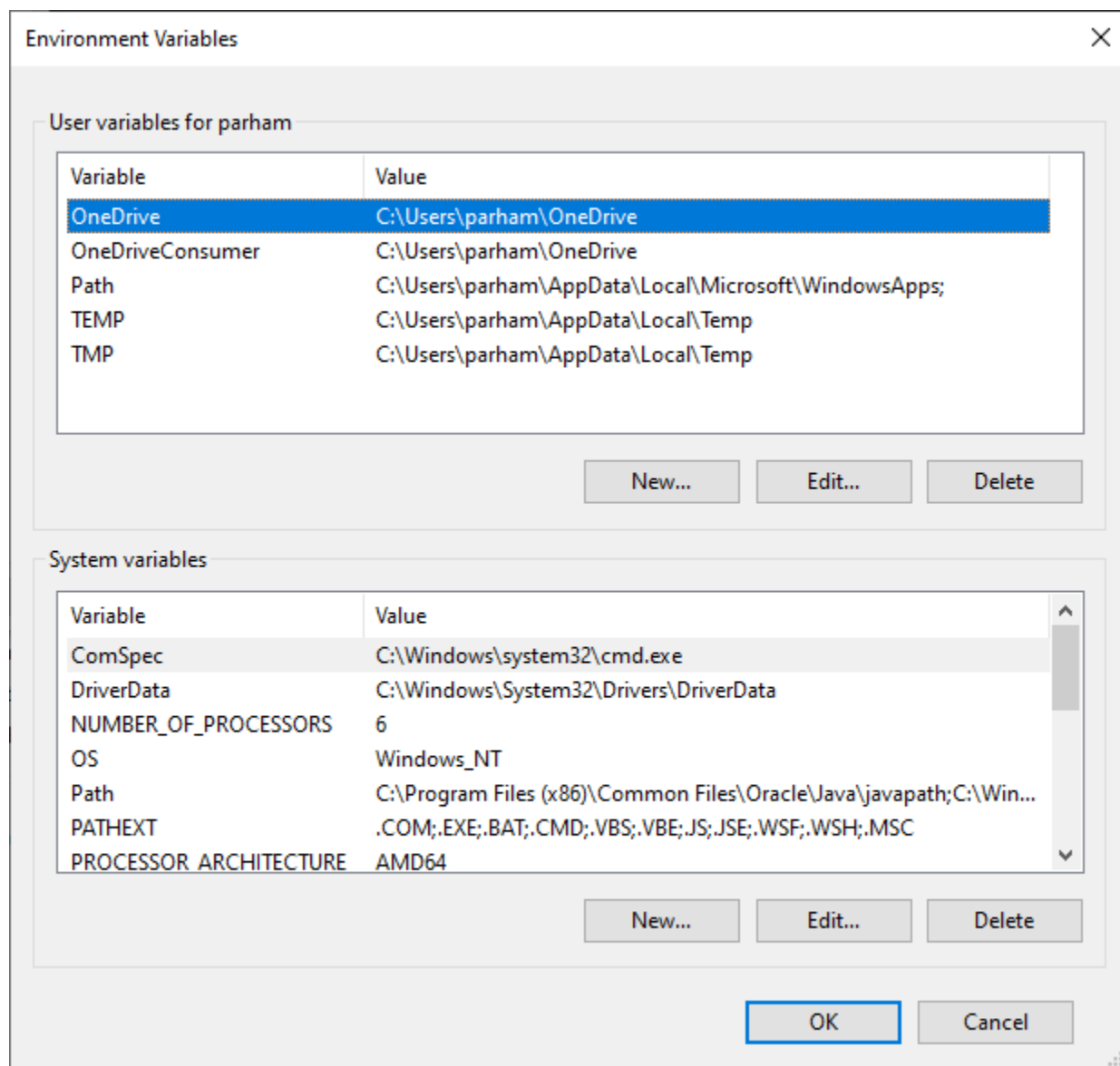
با دستور زیر می‌توانید پوشه را به یک پوشه دیگر تغییر دهید:

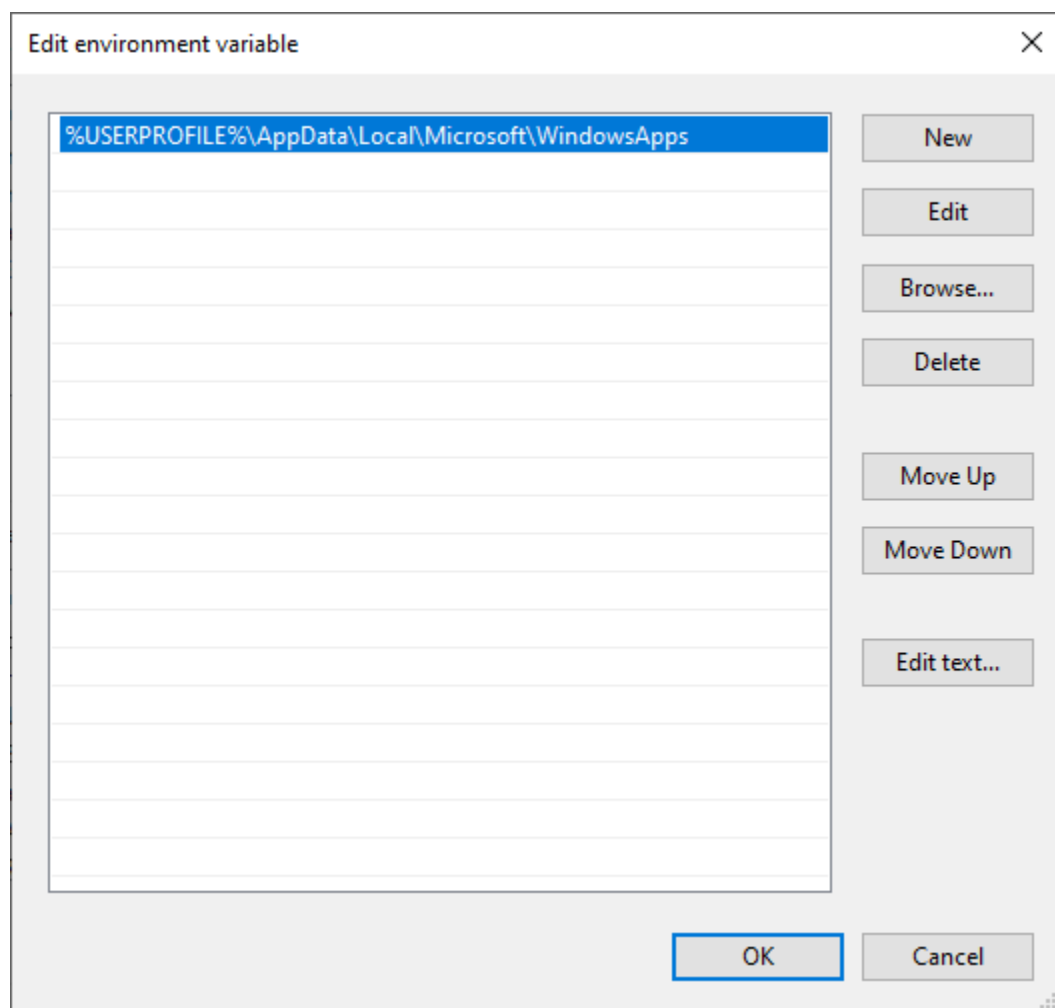
```
cd Downloads
```

```
C:\Users\parham\Downloads>
```

در خط فرمان دستوراتی که فراخوانی می‌شوند در واقع برنامه‌هایی می‌باشند که روی سیستم شما قابل دسترسی می‌باشند. برای هر دستور مسیرهای مشخصی در سیستم شما جستجو می‌شود تا برنامه مورد نظر شما پیدا شود.

این مسیرها در پنجره زیر قابل تنظیم می‌باشند:





در صورتی که می‌خواهید بتوانید از gcc در محیط خط فرمان استفاده کنید، می‌بایست آدرس محل نصب آن را به این متغیر اضافه کنید.

یک برنامه ساده را در فایل hello.c می‌نویسیم:

```
#include <stdio.h>

int main() {
    int n;
    scanf("%d", &n);

    printf("Hello World %d\n", n);
}
```

حال فرض کنید که می‌خواهیم فایل hello.c را که در پوشه فعلی قرار دارد کامپایل کنیم:

```
gcc -o hello.exe hello
```

در ادامه برنامه را اجرا می‌کنیم:

```
hello.exe
```

برای قسمت بعدی قصد داریم مروری بر نرم افزار کنترل نسخه git داشته باشیم. از آنجایی که دانشجویان با خط فرمان آشنا شده‌اند، برای کار با گیت هم از خط فرمان استفاده خواهد شد. برای کنترل نسخه‌های برنامه راه‌های زیادی وجود دارد، یکی از این راه‌ها استفاده از یک نرم افزار کنترل نسخه است. این نرم افزار تغییرات شما، توضیح این تغییرات و اطلاعات دیگری را برای هر تغییر نگهداری می‌کند.

یکی از نرم‌افزارهای کنترل نسخه نرم افزار گیت می‌باشد. این نرم افزار تغییرات شما را در قالب commit ذخیره می‌کند و در نهایت با آن‌ها یک درخت از گذشته پروژه شما می‌سازد که می‌توانید این درخت را پیمایش کرده و در صورت نیاز تغییرات را باز گردانید.

برای شروع یک پروژه روی سایت گیت‌هاب ساخته و آن را با دستور زیر clone کنید:

```
git clone https://github.com/1995parham/hello-wrold
```

دقت داشته باشید که می‌بایست به جای 1995parham از نام کاربری خودتان استفاده کنید. بعد از clone کردن شما یک نسخه از مخزن github خودتان را روی سیستم دارید. حال می‌توانید شروع به نوشتن پروژه کنید. فرض کنید فایل main.c را نوشته و می‌خواهید تغییرات آن را ذخیره کنید.

```
git add main.c
git commit -m "feat: Setup Project"
git push origin master
```

در اینجا منظور از origin مخزن github شما می‌باشد. اگر از دستور push استفاده نکنید، تمامی تغییرات تنها روی سیستم شما موجود خواهد بود. دستورات زیر را در پروژه‌تان امتحان کنید:

```
git status
git log
```

این دستورات وضعیت مخزن محلی شما را نشان می‌دهند و می‌توانید ببینید که در گذشته چه تغییراتی داشتید و وضعیت فعلی فایل‌های شما به چه ترتیبی است.

در نهایت قصد داریم یک پروژه کوچک را روی گیت انجام دهیم. بازی XO را در یک جدول ۳ در ۳ مطابق با روند زیر پیاده‌سازی کنید:

```
Player X:
```

```
1
```

```
X |   |
   |   |
```

```
  |  |  
Player O:
```

```
5
```

```
X |  |
```

```
  | O |
```

```
  |  |
```

```
Player X:
```

```
7
```

```
X |  |
```

```
  | O |
```

```
X |  |
```

```
Player O:
```

```
4
```

```
X |  |
```

```
O | O |
```

```
X |  |
```

```
Player X:
```

```
6
```

```
X |  |
```

```
O | O | X
```

```
X |  |
```

این بازی را پیاده‌سازی کرده و آن را به صورت مرحله به مرحله در گیت‌هاب قرار دهید. در نظر داشته باشید که شما روی سرویس کنترل نسخه نباید هیچ چیز دیگری به جز کدهایتان را قرار دهید پس از قراردادن فایل‌های اضافه مانند خروجی و ... روی نرم افزار کنترل نسخه خودداری کنید.