

- در ابتدا بر اساس مراحل ذکر شده در داک، با وجود مشکلات فراوان اینترنت! ماشین مجازی vagrant را بالا آوردیم.

```
erfan@erfan-R0G-Strix-G512LI:~/ERFAN/ce424-vm$ vagrant ssh
Welcome to Ubuntu 18.04.4 LTS (GNU/Linux 4.15.0-76-generic x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:    https://landscape.canonical.com
 * Support:       https://ubuntu.com/advantage

System information as of Tue Feb 14 15:36:50 UTC 2023

System load:  0.23           Processes:            104
Usage of /:   39.0% of 9.63GB Users logged in:          0
Memory usage: 13%           IP address for enp0s3: 10.0.2.15
Swap usage:   0%            IP address for enp0s8: 192.168.162.162

 * Strictly confined Kubernetes makes edge and IoT secure. Learn how MicroK8s
   just raised the bar for easy, resilient and secure K8s cluster deployment.

https://ubuntu.com/engage/secure-kubernetes-at-the-edge

410 packages can be updated.
347 updates are security updates.

New release '20.04.5 LTS' available.
Run 'do-release-upgrade' to upgrade to it.

Last login: Tue Feb 14 09:04:56 2023 from 10.0.2.2
vagrant@development [15:36:50] ~ $ ls
code
```

- برای دسترسی به فایل‌ها و ادیت کردن آنها از vs code و اکستنشن‌های remote ssh استفاده کردیم. ابتدا به کمک دستور vagrant ssh-config کانفیگ مربوطه را می‌سازیم:

```
Host default
  HostName 127.0.0.1
  User vagrant
  Port 2222
  UserKnownHostsFile /dev/null
  StrictHostKeyChecking no
  PasswordAuthentication no
  IdentityFile /home/erfan/.vagrant.d/boxes/ce424-VAGRANTSLASH-spring2020/1.0.0/virtualbox/vagrant_private_key
  IdentitiesOnly yes
  LogLevel FATAL
  ForwardAgent yes
  ForwardX11 yes
```

- سپس در vs code به کمک این کانفیگ به ماشین دسترسی خواهیم داشت. (لینک استفاده شده:

[https://medium.com/@lopezgand/connect-visual-studio-code-with-vagrant\(-in-your-local-machine-24903fb4a9de](https://medium.com/@lopezgand/connect-visual-studio-code-with-vagrant(-in-your-local-machine-24903fb4a9de)

- سپس برای تکمیل کدهای مربوط به برنامه word_count نیز مطابق داک عمل کردیم. در مورد عملکرد برخی تابع‌ها و تغییرات کدها، به طور خلاصه در زیر توضیحاتی آورده شده است:

word_count.c

len_words:

این تابع تعداد اعضای لیست پیوندی را می‌شمارد (بر روی لیست تا رسیدن به null پیمایش می‌کند)

find_word:

با پیمایش بر روی لیست پیوندی، در صورت پیدا کردن WordCount با word مشابه ورودی، آنرا برمی‌گرداند.

add_word:

این تابع اگر کلمه را پیدا کند (با استفاده از `find_word`)، صرفاً `count` را زیاد می‌کند و در غیر این صورت یک نود از ابتدا به لیست اضافه می‌کند.

main.c

num_words:

این تابع کاراکتر به کاراکتر از فایل می‌خواند، مادامی که کاراکتر `alphabet` باشد و طول آن به ماکسیمم طول یک کلمه نرسید این کار را ادامه می‌دهد. هر گاه یکی از شروط مذکور نقض شود، یک کلمه به تعداد کلمات اضافه می‌شود.

count_words:

این تابع مانند تابع قبل، کاراکتر به کاراکتر می‌خواند و کلمات را تشخیص می‌دهد. یک کار اضافه هم انجام می‌دهد که یک کلمه `word` نگه می‌دارد و پس از تشخیص هر کلمه، آنرا با فراخوانی تابع `add_word` به لیست اضافه می‌کند.

- برای تکمیل فایل `limits.c` از تابع `getrlimit` مطابق منوال لینوکس در <https://man7.org/linux/man-pages/man2/getrlimit.2.html> استفاده کردیم.
 - دو بخش پایانی تمرین بر اساس مراحل آمده در داک انجام شده‌اند و خواسته‌های سوالات در فایل‌های `txt` مربوطه آمده است.
- تصویری از محیط `gdb` در ادامه برای نمونه آمده است:

```
map.c
14 int main(int argc, char *argv[]) {
15     /* A stack allocated variable */
16     volatile int i = 0;
17
18     /* Dynamically allocate some stuff */
19     volatile char *buf1 = malloc(100);
20     /* ... and some more stuff */
21     volatile char *buf2 = malloc(100);
22
23     recur(3);
24     return 0;
}

native process 6285 In: main
(gdb) break 1
Breakpoint 1 at 0x699: file map.c, line 1.
(gdb) run
Starting program: /home/vagrant/code/personal/HW0/map

Breakpoint 1, main (argc=1, argv=0x7fffffffe148) at map.c:16
(gdb) █
```