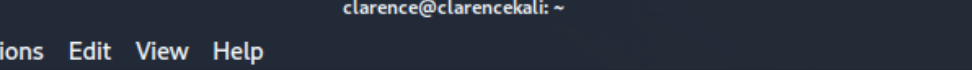


1.2.

3.

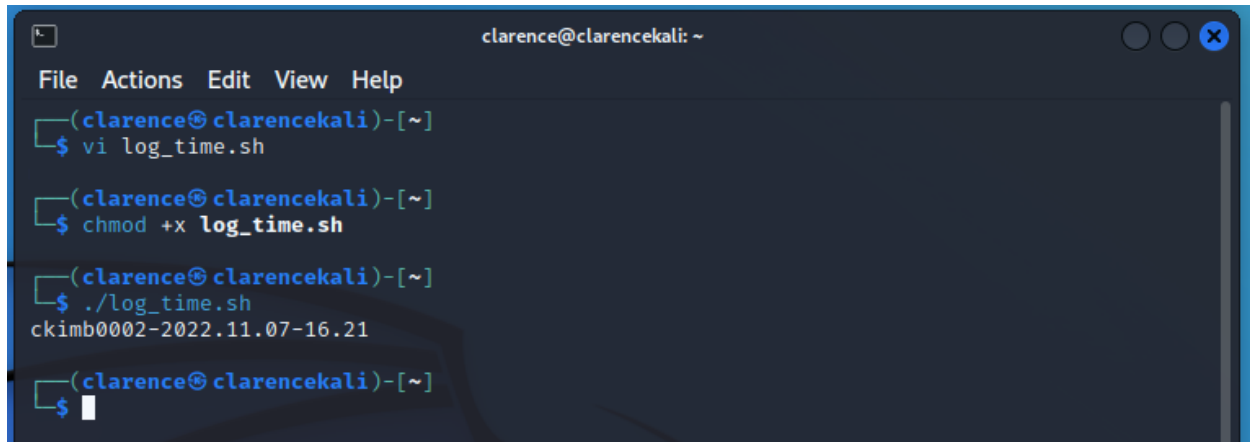
4.



The screenshot shows a terminal window with the title bar "clarence@clarencekali: ~". The menu bar includes "File", "Actions", "Edit", "View", and "Help". The terminal content shows three prompts and commands:

```
(clarence@clarencekali)-[~]  
$ vi log_time.sh  
  
(clarence@clarencekali)-[~]  
$ chmod +x log_time.sh  
  
(clarence@clarencekali)-[~]  
$
```

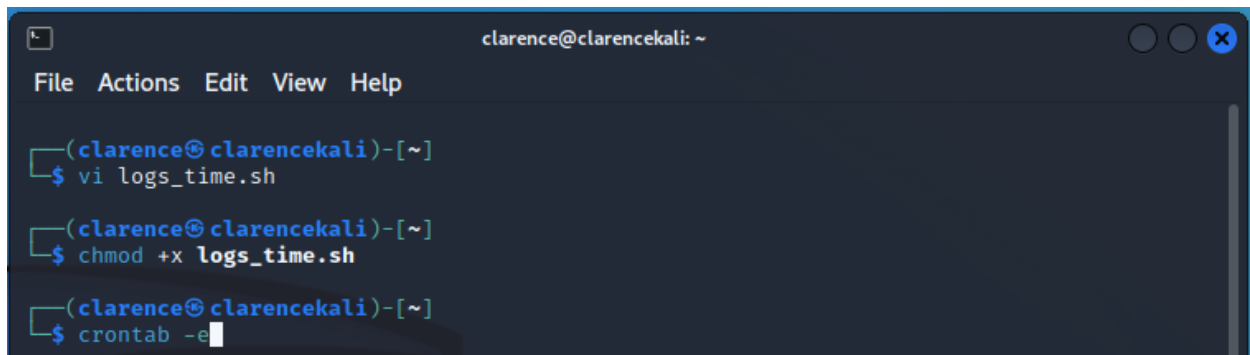
CYSE 270
Clarence V. Kimbrell Jr.
Assignment #9
5.6.



```
clarence@clarencekali: ~  
File Actions Edit View Help  
(clarence@clarencekali)-[~]  
$ vi log_time.sh  
(clarence@clarencekali)-[~]  
$ chmod +x log_time.sh  
(clarence@clarencekali)-[~]  
$ ./log_time.sh  
ckimb0002-2022.11.07-16.21  
(clarence@clarencekali)-[~]  
$
```

Task B

1.2

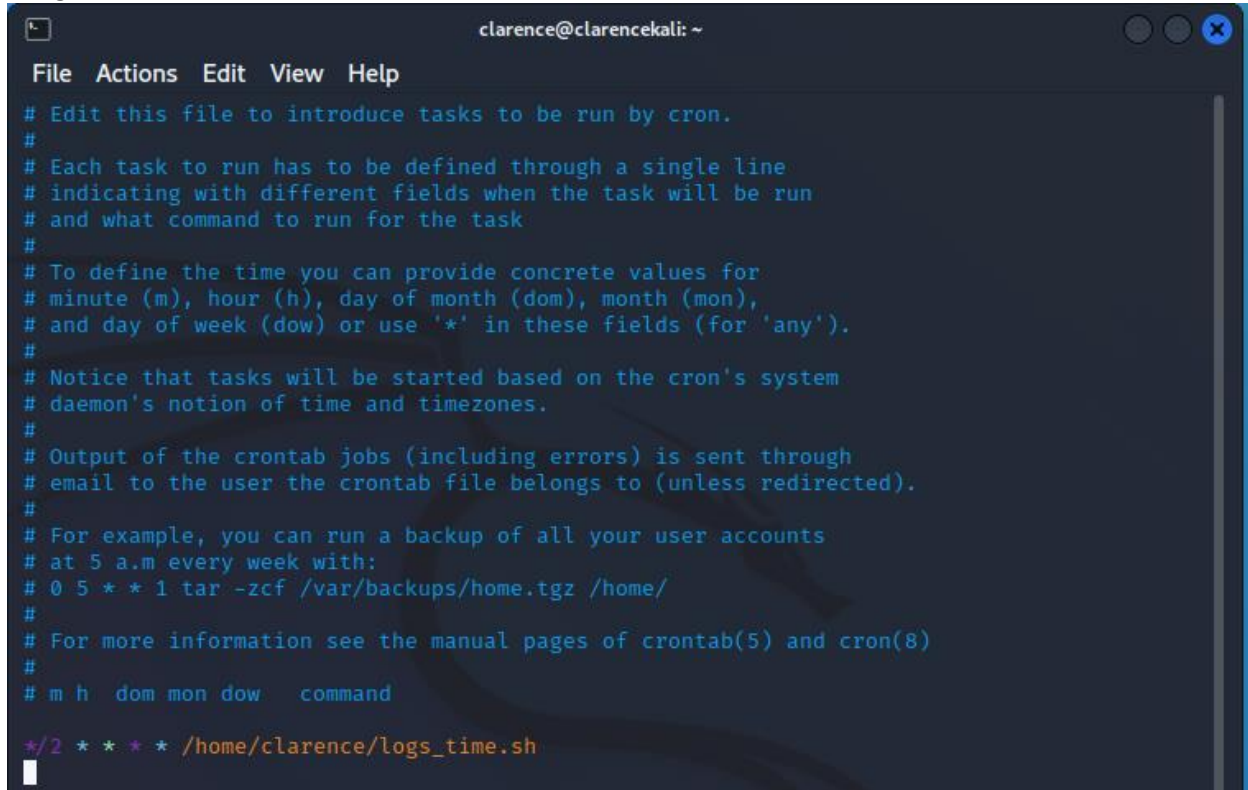


```
clarence@clarencekali: ~  
File Actions Edit View Help  
(clarence@clarencekali)-[~]  
$ vi logs_time.sh  
(clarence@clarencekali)-[~]  
$ chmod +x logs_time.sh  
(clarence@clarencekali)-[~]  
$ crontab -e
```

CYSE 270

Clarence V. Kimbrell Jr.

Assignment #9



```
clarence@clarencekali: ~
File Actions Edit View Help
# Edit this file to introduce tasks to be run by cron.
#
# Each task to run has to be defined through a single line
# indicating with different fields when the task will be run
# and what command to run for the task
#
# To define the time you can provide concrete values for
# minute (m), hour (h), day of month (dom), month (mon),
# and day of week (dow) or use '*' in these fields (for 'any').
#
# Notice that tasks will be started based on the cron's system
# daemon's notion of time and timezones.
#
# Output of the crontab jobs (including errors) is sent through
# email to the user the crontab file belongs to (unless redirected).
#
# For example, you can run a backup of all your user accounts
# at 5 a.m every week with:
# 0 5 * * 1 tar -zcf /var/backups/home.tgz /home/
#
# For more information see the manual pages of crontab(5) and cron(8)
#
# m h dom mon dow  command
*/2 * * * * /home/clarence/logs_time.sh
```

3.

CYSE 270
Clarence V. Kimbrell Jr.
Assignment #9

```
clarence@clarencekali: ~  
File Actions Edit View Help  
# Edit this file to introduce tasks to be run by cron.  
#  
# Each task to run has to be defined through a single line  
# indicating with different fields when the task will be run  
# and what command to run for the task  
#  
# To define the time you can provide concrete values for  
# minute (m), hour (h), day of month (dom), month (mon),  
# and day of week (dow) or use '*' in these fields (for 'any').  
#  
# Notice that tasks will be started based on the cron's system  
# daemon's notion of time and timezones.  
#  
# Output of the crontab jobs (including errors) is sent through  
# email to the user the crontab file belongs to (unless redirected).  
#  
# For example, you can run a backup of all your user accounts  
# at 5 a.m every week with:  
# 0 5 * * 1 tar -zcf /var/backups/home.tgz /home/  
#  
# For more information see the manual pages of crontab(5) and cron(8)  
#  
# m h dom mon dow   command  
*/2 * * * * /home/clarence/logs_time.sh  
~  
~  
~  
~  
~  
~  
:wq
```

4.

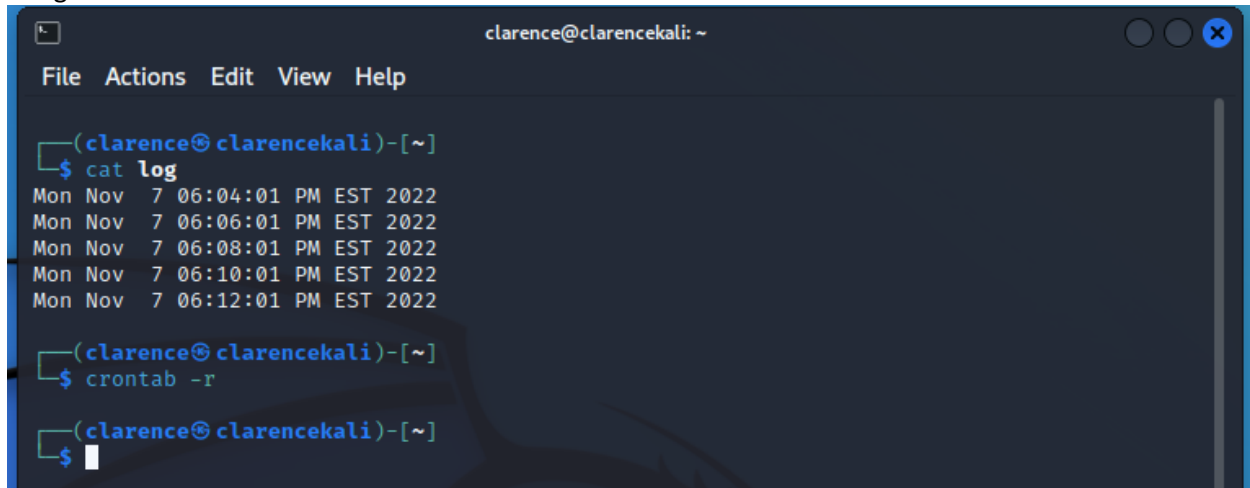
```
clarence@clarencekali: ~  
File Actions Edit View Help  
  
(clarence@clarencekali)-[~]  
$ cat log  
Mon Nov 7 06:04:01 PM EST 2022  
Mon Nov 7 06:06:01 PM EST 2022  
Mon Nov 7 06:08:01 PM EST 2022  
Mon Nov 7 06:10:01 PM EST 2022  
Mon Nov 7 06:12:01 PM EST 2022  
  
(clarence@clarencekali)-[~]  
$
```

5.

CYSE 270

Clarence V. Kimbrell Jr.

Assignment #9



A terminal window titled "clarence@clarencekali: ~" with a menu bar containing "File", "Actions", "Edit", "View", and "Help". The terminal shows the following sequence of commands and output:

```
(clarence@clarencekali)-[~]  
$ cat log  
Mon Nov 7 06:04:01 PM EST 2022  
Mon Nov 7 06:06:01 PM EST 2022  
Mon Nov 7 06:08:01 PM EST 2022  
Mon Nov 7 06:10:01 PM EST 2022  
Mon Nov 7 06:12:01 PM EST 2022  
  
(clarence@clarencekali)-[~]  
$ crontab -r  
  
(clarence@clarencekali)-[~]  
$
```