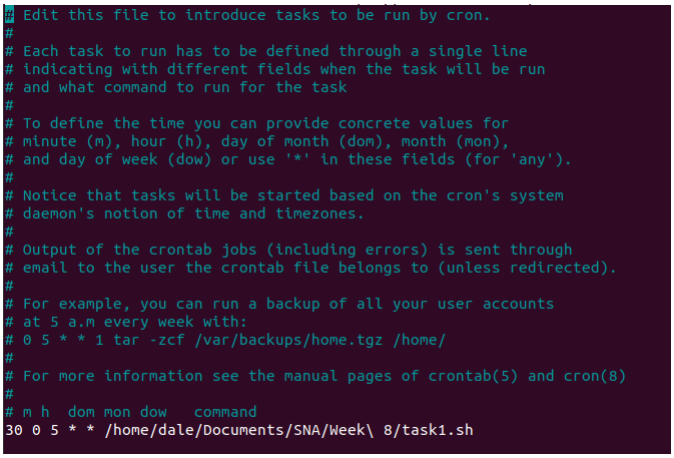
Damir Nabiullin - Lab 8

1. I made such cron job:

   `30 0 5 * * /home/dale/Documents/SNA/Week\ 8/task1.sh`

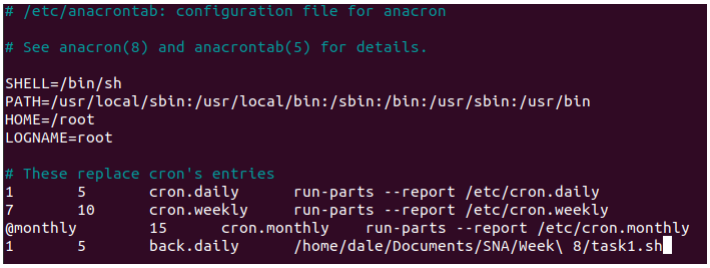
   This job uses such script:

```
#!/bin/bash

to_backup_dir=$"/home/dale/Documents/SNA"
newdir=$"/home/$USER"

if [ ! -d "$newdir" ]
then
    `mkdir $newdir`
fi

if [ -d "$newdir/backup.tar.gz" ]
then
    rm -f $newdir/backup.tar.gz
fi

tar -cvpzf $newdir/backup.tar.gz $to_backup_dir
```



   I made such anacron job:

   `1	5	back.daily	/home/dale/Documents/SNA/Week\ 8/task1.sh`



2. I made such cron:

   `0 0 * * 7 /home/dale/Documents/SNA/Week\ 8/task2.sh`

   And use such backup script:

```
#!/bin/bash

to_backup_dir=$"/var/www/html"
newdir=$"/home/$USER"

if [ ! -d "$newdir" ]
then
    `mkdir $newdir`
fi

if [ -d "$newdir/backup_nginx.tar.gz" ]
then
    rm -f $newdir/backup_nginx.tar.gz
fi

tar -cvpzf $newdir/backup_nginx.tar.gz $to_backup_dir
```

```
# 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
30 0 5 * * /home/dale/Documents/SNA/Week\ 8/task1.sh
0 0 * * 7 /home/dale/Documents/SNA/Week\ 8/task2.sh
```

3. I made such jobs and used such script:

```
#!/bin/bash

dt=$(date '+%d-%m-%Y %H:%M:%S')
echo "$dt $1" >> /var/log/sna_cron.log
```

```
5 0 * * * bash /home/dale/Documents/SNA/Week\ 8/task3.sh "Run five minutes after midnight"
0 10 * * 1-5 bash /home/dale/Documents/SNA/Week\ 8/task3.sh "Run at 10:00 on weekdays"
0 4 * * 1 bash /home/dale/Documents/SNA/Week\ 8/task3.sh "Run at 04:00 every Monday"
0 0 8-14 * 6 bash /home/dale/Documents/SNA/Week\ 8/task3.sh "Run on the second saturday of every monthRun on the second saturday of every month"
```

```
  GNU nano 4.8                                                    /tmp/crontab.1HKYSR/crontab
# 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
30 0 5 * * /home/dale/Documents/SNA/Week\ 8/task1.sh
0 0 * * 7 /home/dale/Documents/SNA/Week\ 8/task2.sh
5 0 * * * bash /home/dale/Documents/SNA/Week\ 8/task3.sh "Run five minutes after midnight"
0 10 * * 1-5 bash /home/dale/Documents/SNA/Week\ 8/task3.sh "Run at 10:00 on weekdays"
0 4 * * 1 bash /home/dale/Documents/SNA/Week\ 8/task3.sh "Run at 04:00 every Monday"
0 0 8-14 * 6 bash /home/dale/Documents/SNA/Week\ 8/task3.sh "Run on the second saturday of every monthRun on the second saturday of every month"
```