```
joe@resources-thegummibear:~$ crontab -l
no crontab for joe
joe@resources-thegummibear:~$
```
joe@resources-thegummibear:~$ EDITOR=vi crontab -e
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).
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
```

```
13 13 * 4,9 1-5 /usr/necl/bin/cpuhog -s 1000
"crontab.nx5DR1/crontab" 24L, 934C written 24,44
```
joe@resouces-thegummibear:~$ crontab -l
# 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

13 13 * 4,9 1-5 /usr/locl/bin/cpuhog -s 1000
sudo grep CRON /var/log/syslog
Sep 19 07:17:01 resources CRON[3905]: (root) CMD ( cd / && run-parts --report /etc/cron.hourly)
Sep 19 08:17:01 resources CRON[3908]: (root) CMD ( cd / && run-parts --report /etc/cron.hourly)
Sep 19 09:17:01 resources CRON[3919]: (root) CMD ( cd / && run-parts --report /etc/cron.hourly)

joe@resources-thegummibear: