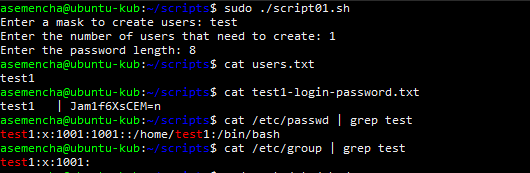
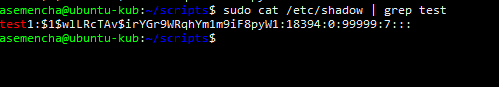
1. Create an automatic user password generator. Users.txt login file that uses a list of users

Automatically generate passwords for all users, create a linux-system with generated passwords in the user’s script so that you can log in with the name of each user with a password created for it (here it is supposed to work with openssl), and create a user \* file for each user -login-password.txt, in which to put the username and the generated password.

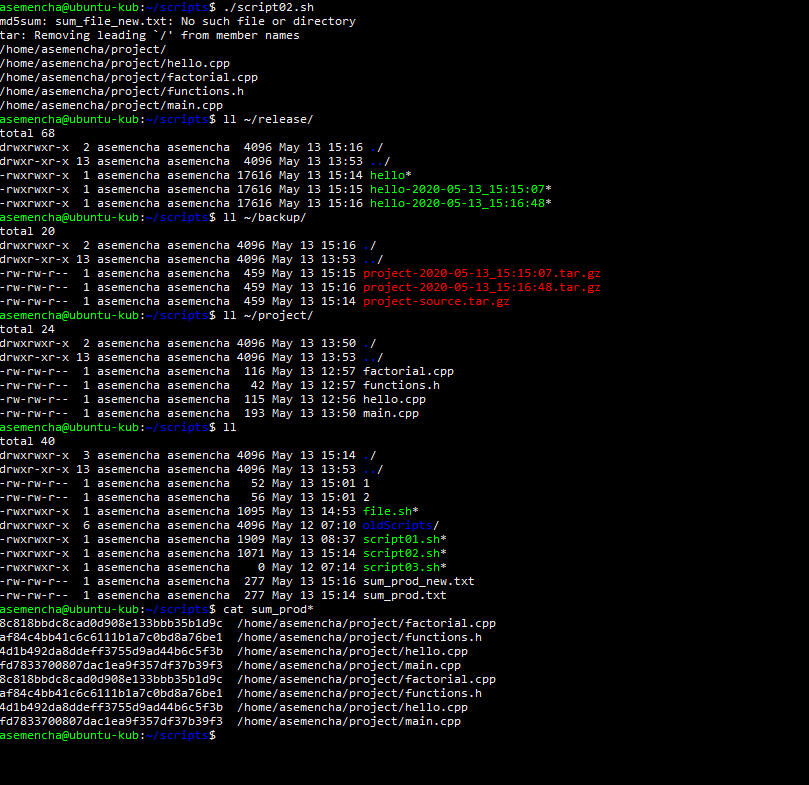




1. Take as a basis the project https://habr.com/en/post/155201/.

Write a script that does the following:

* 1. At the initial start-up - calculation of checksums and archiving of the project.
  2. periodic review of the project for changes.
  3. If the project has changed, then write the new version to a new archive and start recompiling the project.



1. Create a script to collect system statistics

At the initialization stage:

* Create a task for cron, according to which every 5 minutes the file ~/memory/stat, as well as the logs received by the main script, will be packed into the archive.
* Create a task for cron, according to which every 2 minutes information about the current state of memory will be added to the ~/memory/stat file, without considering the size of the swap and header.

The main stage is performed every two minutes:

* Using the vmstat command, for 30 s with an interval of 3 s, collect statistics on the use of system resources.
* Calculate the average number of kernel context switches per second over a given time interval. Information in the log.
* Get information about the average processor over the past 15s. Information in the log.
* Describe the current status of the memory pages available on your system. Information in the log.
* Describe the current status of the hard disk partitions available on your system. Information in the log.

Unfortunately I did not have time to perform this task.