

Module 4 Linux Essentials with Bash (Task 4.12)

4.12.1 Создать автоматический генератор паролей пользователей. На вход скрипта подать файл users.txt в котором содержится список пользователей:

```
1. user1
2. user2
...
20. user20
```

Автоматически сгенерировать пароли для всех пользователей, создать в скрипте пользователей linux-системы со сгенерированными паролями, так чтобы вы могли войти под именем каждого из пользователей с созданным для него паролем (тут предполагается работа с openssl), а также создать для каждого пользователя файл user*-login-password.txt, в который поместить имя пользователя и сгенерированный пароль.

Например:

user1 – uR44y6!#

upassgen.sh (комментарии в коде) - в качестве параметра при запуске передаётся файл в указанном формате. После первого запуска скрипта удалил часть пользователей, чтоб показать работу скрипта при попытке создать пользователя с существующим в системе логином

```
user2@wibob-X61: ~ (ssh) %1 bruh@wibob-X61: ~/devops/scripts_4_12 (ssh) %2
bruh@wibob-X61:~/devops/scripts_4_12$ ./upassgen.sh userlist.txt
Only root may add a user to the system.
bruh@wibob-X61:~/devops/scripts_4_12$ sudo ./upassgen.sh userlist.txt
User user1 has been added to system!
User user2 has been added to system!
User user3 has been added to system!
User user4 has been added to system!
User user5 has been added to system!
User user6 has been added to system!
user7 exists!
user8 exists!
user9 exists!
user10 exists!
user11 exists!
user12 exists!
user13 exists!
user14 exists!
user15 exists!
user16 exists!
user17 exists!
user18 exists!
user19 exists!
user20 exists!
bruh@wibob-X61:~/devops/scripts_4_12$ cat user2-loginpassword.txt
user2 - Hx9587RR
bruh@wibob-X61:~/devops/scripts_4_12$ su user2
Password:
user2@wibob-X61:/home/bruh/devops/scripts_4_12$ cd ~
user2@wibob-X61:~$ pwd
/home/user2
user2@wibob-X61:~$
```

4.12.2 Взять за основу проект <https://habr.com/ru/post/155201/> . Написать скрипт выполняющий следующее:

1. При первоначальном запуске – вычисление контрольных сумм и архивация проекта.
2. периодическая проверка проекта на предмет изменений.
3. Если проект изменился, то записать новую версию в новый архив и запустить перекомпиляцию проекта.

все файлы и каталоги находятся в отдельной директории `c_project`.
в корне этого каталога лежит сам скрипт `builder.sh` в файле `project_files` перечисляются файлы, которые нужно отслеживать (я включил также `Makefile`) актуальные контрольные суммы файлов входящих текущий билд с датой их последнего обновления в билде хранятся в файле `cur_checksum`. Все исходники в том числе `'Makefile'` хранятся в подкаталоге `/src` архивы файлов (в имени дата помещения в архив) хранятся в подкаталоге `/arch`, а откомпилированный бинарник в каталоге `/bin` (остальные комментарии в коде скрипта) Периодическую проверку на предмет изменений логичней всего реализовать с помощью `cron`.

при очищенном файле `cur_checksum` и запуске скрипта, происходит архивирование исходников и запуск компилятора, на выходе архивы (старые специально не удалялись), бинарник, файл реестра

```
bruha@wibob-X61:~/devops/scripts_4_12/c_project/arch$ cd ../; cat cur_checksum
fd7833700807dac1ea9f357df37b39f3 ./src/main.cpp понеділок, 11 травня 2020 09:22:33 +0300
4d1b492da8ddef3755d9ad44b6c5f3b ./src/hello.cpp понеділок, 11 травня 2020 09:22:33 +0300
8c818bbdc8cad0d908e133bbb35b1d9c ./src/factorial.cpp понеділок, 11 травня 2020 09:22:33 +0300
af84c4bb41c6c611b1a7c0bd8a76be1 ./src/functions.h понеділок, 11 травня 2020 09:22:33 +0300
f0aa44405ba6658db7e6f855f9a5eb7a ./src/Makefile понеділок, 11 травня 2020 09:22:33 +0300
bruha@wibob-X61:~/devops/scripts_4_12/c_project$ ls ./bin/
hello
bruha@wibob-X61:~/devops/scripts_4_12/c_project$ hello

Command 'hello' not found, but can be installed with:

sudo snap install hello          # version 2.10, or
sudo apt install hello
sudo apt install hello-traditional

See 'snap info hello' for additional versions.

bruha@wibob-X61:~/devops/scripts_4_12/c_project$ ./hello
bash: ./hello: No such file or directory
bruha@wibob-X61:~/devops/scripts_4_12/c_project$ ./bin/hello
Hello World!
The factorial of 5 is 120
bruha@wibob-X61:~/devops/scripts_4_12/c_project$ ll ./arch/
total 76
drwxrwxr-x 2 bruha bruha 4096 тра 11 09:22 ./
drwxrwxr-x 5 bruha bruha 4096 тра 11 09:22 ../
-rw-rw-r-- 1 bruha bruha 167 тра 11 07:57 Makefile-1-2020-05-11-07-57-36.tgz
-rw-rw-r-- 1 bruha bruha 166 тра 11 07:59 Makefile-1-2020-05-11-07-59-40.tgz
-rw-rw-r-- 1 bruha bruha 166 тра 11 08:11 Makefile-1-2020-05-11-08-11-12.tgz
-rw-rw-r-- 1 bruha bruha 166 тра 11 08:19 Makefile-1-2020-05-11-08-19-05.tgz
-rw-rw-r-- 1 bruha bruha 166 тра 11 08:27 Makefile-1-2020-05-11-08-27-09.tgz
-rw-rw-r-- 1 bruha bruha 167 тра 11 08:28 Makefile-1-2020-05-11-08-28-35.tgz
-rw-rw-r-- 1 bruha bruha 173 тра 11 08:53 Makefile-2020-05-11-08-53-41.tgz
-rw-rw-r-- 1 bruha bruha 173 тра 11 08:58 Makefile-2020-05-11-08-58-38.tgz
-rw-rw-r-- 1 bruha bruha 173 тра 11 09:22 Makefile-2020-05-11-09-22-33.tgz
-rw-rw-r-- 1 bruha bruha 213 тра 11 07:57 factorial.cpp-2020-05-11-07-57-36.tgz
-rw-rw-r-- 1 bruha bruha 213 тра 11 09:22 factorial.cpp-2020-05-11-09-22-33.tgz
-rw-rw-r-- 1 bruha bruha 164 тра 11 07:57 functions.h-2020-05-11-07-57-36.tgz
-rw-rw-r-- 1 bruha bruha 164 тра 11 09:22 functions.h-2020-05-11-09-22-33.tgz
-rw-rw-r-- 1 bruha bruha 225 тра 11 07:57 hello.cpp-2020-05-11-07-57-36.tgz
-rw-rw-r-- 1 bruha bruha 225 тра 11 09:22 hello.cpp-2020-05-11-09-22-33.tgz
-rw-rw-r-- 1 bruha bruha 252 тра 11 07:57 main.cpp-2020-05-11-07-57-36.tgz
-rw-rw-r-- 1 bruha bruha 252 тра 11 09:22 main.cpp-2020-05-11-09-22-33.tgz
bruha@wibob-X61:~/devops/scripts_4_12/c_project$
```

для примера заменим в Makefile имя бинарника. Как видно в файле реестра изменились данные по файлу 'Makefile', в подкаталоге /bin появился новый исполняемый файл и в каталоге /arch со временем создания 9:31 появился новый архив

```
bru@wibob-X61:~/devops/scripts_4_12/c_project$ ./builder.sh
./src/Makefile
g++ main.cpp hello.cpp factorial.cpp -o ../bin/hello_world
bru@wibob-X61:~/devops/scripts_4_12/c_project$ cat cur_checksum
fd7833700807dac1ea9f357df37b39f3 ./src/main.cpp понеділок, 11 травня 2020 09:22:33 +0300
4d1b492da8ddef3755d9ad44b6c5f3b ./src/hello.cpp понеділок, 11 травня 2020 09:22:33 +0300
8c818bbdc8cad0d908e133bbb35b1d9c ./src/factorial.cpp понеділок, 11 травня 2020 09:22:33 +0300
af84c4bb41c6c611b1a7c0bd8a76be1 ./src/functions.h понеділок, 11 травня 2020 09:22:33 +0300
3a4f4367e608db1d12b362de9ace5f5b ./src/Makefile понеділок, 11 травня 2020 09:31:51 +0300
bru@wibob-X61:~/devops/scripts_4_12/c_project$ ls ll ./bin/
ls: cannot access 'll': No such file or directory
./bin/:
hello hello_world
bru@wibob-X61:~/devops/scripts_4_12/c_project$ ll ./bin/
total 24
drwxrwxr-x 2 bru bru 4096 тра 11 09:31 ./
drwxrwxr-x 5 bru bru 4096 тра 11 09:31 ../
-rwxrwxr-x 1 bru bru 8000 тра 11 09:22 hello*
-rwxrwxr-x 1 bru bru 8000 тра 11 09:31 hello_world*
bru@wibob-X61:~/devops/scripts_4_12/c_project$ ll ./arch/
total 80
drwxrwxr-x 2 bru bru 4096 тра 11 09:31 ./
drwxrwxr-x 5 bru bru 4096 тра 11 09:31 ../
-rw-rw-r-- 1 bru bru 167 тра 11 07:57 Makefile-1-2020-05-11-07-57-36.tgz
-rw-rw-r-- 1 bru bru 166 тра 11 07:59 Makefile-1-2020-05-11-07-59-40.tgz
-rw-rw-r-- 1 bru bru 166 тра 11 08:11 Makefile-1-2020-05-11-08-11-12.tgz
-rw-rw-r-- 1 bru bru 166 тра 11 08:19 Makefile-1-2020-05-11-08-19-05.tgz
-rw-rw-r-- 1 bru bru 166 тра 11 08:27 Makefile-1-2020-05-11-08-27-09.tgz
-rw-rw-r-- 1 bru bru 167 тра 11 08:28 Makefile-1-2020-05-11-08-28-35.tgz
-rw-rw-r-- 1 bru bru 173 тра 11 08:53 Makefile-2020-05-11-08-53-41.tgz
-rw-rw-r-- 1 bru bru 173 тра 11 08:58 Makefile-2020-05-11-08-58-38.tgz
-rw-rw-r-- 1 bru bru 173 тра 11 09:22 Makefile-2020-05-11-09-22-33.tgz
-rw-rw-r-- 1 bru bru 179 тра 11 09:31 Makefile-2020-05-11-09-31-51.tgz
-rw-rw-r-- 1 bru bru 213 тра 11 07:57 factorial.cpp-2020-05-11-07-57-36.tgz
-rw-rw-r-- 1 bru bru 213 тра 11 09:22 factorial.cpp-2020-05-11-09-22-33.tgz
-rw-rw-r-- 1 bru bru 164 тра 11 07:57 functions.h-2020-05-11-07-57-36.tgz
-rw-rw-r-- 1 bru bru 164 тра 11 09:22 functions.h-2020-05-11-09-22-33.tgz
-rw-rw-r-- 1 bru bru 225 тра 11 07:57 hello.cpp-2020-05-11-07-57-36.tgz
-rw-rw-r-- 1 bru bru 225 тра 11 09:22 hello.cpp-2020-05-11-09-22-33.tgz
-rw-rw-r-- 1 bru bru 252 тра 11 07:57 main.cpp-2020-05-11-07-57-36.tgz
-rw-rw-r-- 1 bru bru 252 тра 11 09:22 main.cpp-2020-05-11-09-22-33.tgz
bru@wibob-X61:~/devops/scripts_4_12/c_project$
```

при повторных запусках скрипта, но при отсутствии изменений в отслеживаемых файлах ничего не происходит.

4.12.3 Создать скрипт сбора статистики работы системы

На этапе инициализации:

Создайте задание для cron, согласно которому каждые 5 минут файл ~/memory/stat, а также логи, полученные основным скриптом, будут упаковываться в архив.

Создайте задание для cron, согласно которому каждые 2 минуты в файл ~/memory/stat будет добавляться информация о текущем состоянии памяти, без учета размера подкачки и заголовка.

Основной этап выполнять каждые две минуты:

С помощью команды vmstat, в течении 30с с интервалом в 3с, собирайте статистику об использовании ресурсов системы. Посчитайте среднее количество переключений контекста ядра в секунду на заданном интервале времени. Информацию – в лог.

Получите информацию о средней загрузженности процессора в течении последних 15м.

Информацию – в лог.

Опишите текущее состояние страниц памяти, доступных в вашей системе. Информацию – в лог.

Опишите текущее состояние разделов жестких дисков, доступных в вашей системе.
Информацию – в лог.

структура и расположение файлов к заданию:

- `memory_stat.sh` пишет в `~/memory/stat` каждые 2 минуты
- `sys_avg_stat.sh` пишет в `~/devops/scripts_4_12/system_avg_stat` каждые 2 минуты
- `stat_backup.sh` каждые 5 минут архивирует верхние два в отдельные файлы (в имени маркер времени) в каталоге `~/devops/scripts_4_12/backup`

скрин `crontab`

```
# For more information see the manual pages of crontab(5) and cron(8)
#
# m h dom mon dow   command
*/2 * * * * /home/bruha/devops/scripts_4_12/memory_stat.sh
*/5 * * * * /home/bruha/devops/scripts_4_12/stat_backup.sh
*/2 * * * * /home/bruha/devops/scripts_4_12/sys_avg_stat.sh
# * * * * * * echo "ku!" >>~/memory/stat
```

содержимое файла статистики по памяти

bruha@wibob-X61: ~/devops/scripts_4_12 (ssh)					bruha@wibob-X61: /var/log (ssh)				
%1					%2				
2504104	488036	216788	130804	1799280	1583352	2020-05-11	15:12:01		
2504104	487932	216880	130804	1799292	1583456	2020-05-11	15:14:01		
2504104	493028	211748	130804	1799328	1578360	2020-05-11	15:16:01		
2504104	489664	215108	130804	1799332	1581724	2020-05-11	15:18:01		
2504104	486632	218144	130804	1799328	1584756	2020-05-11	15:20:01		
2504104	486008	218756	130804	1799340	1585380	2020-05-11	15:22:01		
2504104	485476	219276	130804	1799352	1585912	2020-05-11	15:24:01		
2504104	485656	219104	130804	1799344	1585732	2020-05-11	15:26:01		
2504104	485796	218956	130804	1799352	1585592	2020-05-11	15:28:01		
2504104	486748	217988	130804	1799368	1584640	2020-05-11	15:30:01		
2504104	485988	218732	130804	1799384	1585400	2020-05-11	15:32:01		
2504104	485728	218980	130804	1799396	1585660	2020-05-11	15:34:01		
2504104	485808	218752	130804	1799544	1585540	2020-05-11	15:36:01		
2504104	485524	219032	130804	1799548	1585864	2020-05-11	15:38:01		
2504104	488848	215668	130804	1799588	1582536	2020-05-11	15:40:01		

содержимое файла `system_avg_stat`

```
HugePages_Surp:      0
Hugepagesize:       2048 kB
DirectMap4k:        12280 kB
DirectMap2M:        894976 kB
-----
cs_avg: 172.455
-----
15min load average: 0.00
-----
MemTotal:           2504104 kB
MemFree:            216400 kB
MemAvailable:       1583680 kB
Buffers:            336456 kB
Cached:             1224316 kB
SwapCached:         0 kB
Active:             1262024 kB
Inactive:           661872 kB
Active(anon):       366784 kB
Inactive(anon):     129564 kB
Active(file):       895240 kB
Inactive(file):     532308 kB
Unevictable:        3428 kB
Mlocked:            3428 kB
HighTotal:          1704776 kB
HighFree:           155604 kB
LowTotal:           790328 kB
LowFree:            60796 kB
SwapTotal:          0 kB
SwapFree:           0 kB
Dirty:              16 kB
```

содержимое каталога *backup*

```
bru@wibob-X61:~/devops/scripts_4_12$ ls ./backup/
stat_on_2020-05-11-12:20:20.tgz  stat_on_2020-05-11-13:25:01.tgz  stat_on_2020-05-11-14:15:01.tgz  stat_on_2020-05-11-15:05:01.tgz  stat_on_2020-05-11-15:55:01.tgz
stat_on_2020-05-11-12:21:49.tgz  stat_on_2020-05-11-13:30:01.tgz  stat_on_2020-05-11-14:20:01.tgz  stat_on_2020-05-11-15:10:01.tgz  stat_on_2020-05-11-16:00:01.tgz
stat_on_2020-05-11-12:35:01.tgz  stat_on_2020-05-11-13:35:02.tgz  stat_on_2020-05-11-14:25:01.tgz  stat_on_2020-05-11-15:15:01.tgz  stat_on_2020-05-11-16:05:01.tgz
stat_on_2020-05-11-12:50:01.tgz  stat_on_2020-05-11-13:40:01.tgz  stat_on_2020-05-11-14:30:01.tgz  stat_on_2020-05-11-15:20:01.tgz  stat_on_2020-05-11-16:10:01.tgz
stat_on_2020-05-11-12:55:01.tgz  stat_on_2020-05-11-13:45:01.tgz  stat_on_2020-05-11-14:35:01.tgz  stat_on_2020-05-11-15:25:01.tgz  stat_on_2020-05-11-16:15:01.tgz
stat_on_2020-05-11-13:00:02.tgz  stat_on_2020-05-11-13:50:01.tgz  stat_on_2020-05-11-14:40:01.tgz  stat_on_2020-05-11-15:30:01.tgz  stat_on_2020-05-11-16:20:01.tgz
stat_on_2020-05-11-13:05:01.tgz  stat_on_2020-05-11-13:55:01.tgz  stat_on_2020-05-11-14:45:01.tgz  stat_on_2020-05-11-15:35:01.tgz  stat_on_2020-05-11-16:25:01.tgz
stat_on_2020-05-11-13:10:01.tgz  stat_on_2020-05-11-14:00:01.tgz  stat_on_2020-05-11-14:50:01.tgz  stat_on_2020-05-11-15:40:01.tgz  stat_on_2020-05-11-16:30:01.tgz
stat_on_2020-05-11-13:15:01.tgz  stat_on_2020-05-11-14:05:01.tgz  stat_on_2020-05-11-14:55:01.tgz  stat_on_2020-05-11-15:45:01.tgz  stat_on_2020-05-11-16:35:01.tgz
stat_on_2020-05-11-13:20:01.tgz  stat_on_2020-05-11-14:10:02.tgz  stat_on_2020-05-11-15:00:01.tgz  stat_on_2020-05-11-15:50:01.tgz  stat_on_2020-05-11-16:40:01.tgz
bru@wibob-X61:~/devops/scripts_4_12$
```

листинг *memory_stat.sh*

```
set -o nounset # Treat unset variables as an error
free | awk -v date="$(date +"%F %T")" 'FNR == 2 { printf "%14s %12s %12s %12s %12s %12s %s\n", $2, $3, $4, $5, $6, $7, date}' >>~/memory/stat
```

листинг *sys_avg_stat.sh*

```
set -o nounset # Treat unset variables as an error

echo "===== " >> ~/devops/scripts_4_12/system_avg_stat
date +%D %R" >> system_avg_stat
echo "-----" >> ~/devops/scripts_4_12/system_avg_stat
vmstat --one 3 12 | awk 'FNR>3 {cs_avg += $12; n++; } END { if (n > 0) print "cs_avg: " cs_avg / n; }' >> ~/devops/scripts_4_12/system_avg_stat
## на 30 секундах начиная с нулевой отметки помещается 11 измерений, первое отбрасываем так как не релевантны данные
## поэтому в расчет среднего идут 11 строк из 12 снятых (1 строка с донными игнорируется)
echo "-----" >> ~/devops/scripts_4_12/system_avg_stat
echo "15min load average: `uptime | cut -d " " --fields=15`" >> ~/devops/scripts_4_12/system_avg_stat
echo "-----" >> ~/devops/scripts_4_12/system_avg_stat
cat /proc/meminfo >> ~/devops/scripts_4_12/system_avg_stat
echo "-----" >> ~/devops/scripts_4_12/system_avg_stat
sudo fdisk -l >> ~/devops/scripts_4_12/system_avg_stat
echo "-----" >> ~/devops/scripts_4_12/system_avg_stat
```

листинг *stat_backup.sh*

```
set -o nounset # Treat unset variables as an error
tar zcf /home/bru/devops/scripts_4_12/backup/stat_on_$(date +"%F-%T").tgz ~/memory/stat ~/devops/scripts_4_12/system_avg_stat
```