

Ćwiczenia 4 – Ubuntu serwer – crontab, at

Zaloguj się na swoje konto imienXYZ, gdzie XYZ oznacza kod klasy i grupy, np. jank3t1
Jeśli nie masz konta, **sudo adduser** imienXYZ

1. Dodaj swoje konto do grupy sudo: **sudo usermod twoje_konto -G sudo**
2. Sprawdzenie czy jesteśmy w grupie sudo: **id konto**
3. Wyдай komendę na piątym terminalu: **man at**
4. Wyдай komendę na szóstym terminalu: **info crontab**
5. Otwórz pomocniczo na czwartym terminalu: **cat /etc/crontab**
6. Otwórz stronę pomocy, np.: <https://crontab.guru/>
7. Zainstaluj obsługę poczty: **sudo apt install mailutils** podaj tylko lokanie
8. Sprawdzić pocztę komendą **mail**
9. Z pomocą polecenia crontab zaplanuj:

- a) Archiwizuj swój katalog domowy co 1 minutę do katalogu /var/backups oraz /tmp z użyciem tar i gzipa

```
andrzej@servubu:~$ crontab -l
# kopia mojego katalogu domowego
*/1 * * * * tar cfz /var/backups/home-andrzej.tar.gz -P /home/andrzej
*/1 * * * * tar cfz /tmp/home-andrzej.tar.gz -P /home/andrzej
```

- b) Utwórz co 1 minutę podając dzień i miesiąc plik o nazwie plikcrontabtest w ~

```
# tworzenie pliku testowego
```

```
*/1 * 12 1 * touch /home/andrzej/plikcrontabtest
```

- c) Uruchom skrypt co 1 minutę w każdy dzień roboczy pobierający wycenę bitcoina ze strony:

<https://api.coinbase.com>

Przygotowanie, test curl i jq

```
andrzej@servubu:~$ curl https://api.coinbase.com/v2/prices/spot?currency=USD
{"data":{"base":"BTC","currency":"USD","amount":"18228.06"}}andrzej@servubu:~$
andrzej@servubu:~$ jason=$(curl https://api.coinbase.com/v2/prices/spot?currency=USD)
% Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
           %             %             Dload  Upload  Total   Spent    Left   Speed
100    60      0    60      0      0   1153      0  --:--:--  --:--:--  --:--:--   1176
andrzej@servubu:~$ echo $jason
{"data":{"base":"BTC","currency":"USD","amount":"18228.06"}}
andrzej@servubu:~$ jq -r '.data.amount' <<< $jason
18228.06
-
```

Przykładowa realizacja:

```
Open  skrypt2.sh  Save
1 #!/bin/bash
2
3 jason=$(curl https://api.coinbase.com/v2/prices/spot?currency=USD)
4 sleep 2
5 cena_bitcoina=$(jq -r '.data.amount' <<< ${jason})
6 echo "Cena bitcoina: $cena_bitcoina USD w dniu $(date)" >> $HOME/cena_bitcoina.txt
```

Zadanie crona:

```
andrzej@servubu:~$ crontab -l
# kopia mojego katalogu domowego
*/1 * * * * tar cfz /tmp/home-andrzej.tar.gz -P /home/andrzej

# tworzenie pliku tekstowego

*/1 * 12 1 * touch /home/andrzej/plikcrontabtest

# skrypt sprawdzajacy kurs waluty

*/1 * * * mon-fri $HOME/skrypt2.sh
```

Efekt działania:

```
andrzej@servubu:~$ cat cena_bitcoina.txt
Cena bitcoina: 16816.78 USD w dniu Wed 04 Jan 2023 03:58:03 PM CET
Cena bitcoina: 16813.76 USD w dniu Wed 04 Jan 2023 03:59:04 PM CET
```

- d) uruchomienie skryptu skrypt3.sh zbierającego statystyki pracy karty sieciowej z ostatnich 5 minut o zadanej godzinie, ale tylko w środy i piątki oraz w dniu ćwiczeń

```
andrzej@servubu:~$ sudo apt install vnstat vnstat -y
Reading package lists... Done
```

```
Open  skrypt3.sh
1 #!/bin/bash
2
3 /usr/bin/vnstat -5 > $HOME/stat_net.txt
```

Zadanie crona:

```
# statystyki sieci co 5 minut

*/1 * * * 3-5 $HOME/skrypt3.sh
```

Sprawdzenie:

```
andrzej@servubu:~$ cat stat_net.txt
```

```
enp0s3 / 5 minute
```

time	rx	tx	total	avg. rate
-----+-----+-----+-----				
2023-01-12				
11:55	253.73 KiB	20.12 KiB	273.85 KiB	7.48 kbit/s
12:00	35.05 KiB	8.75 KiB	43.81 KiB	1.20 kbit/s

- e) uruchomienie skryptu skrypt4.sh zbierającego statystyki pracy karty sieciowej do pliku graficznego z ostatnich 5 minut o zadanej godzinie, ale tylko w środy i piątki oraz w dniu ćwiczeń

Zawartość skryptu:

```

1 #!/bin/bash
2
3 /usr/bin/vnstat -5 -o $HOME/stat_net.png
4

```

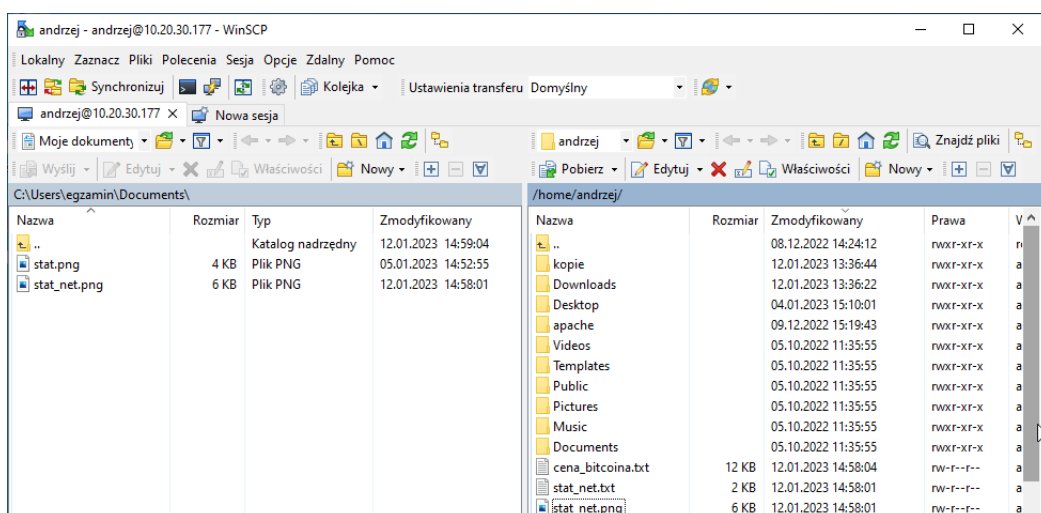
Zadanie crona:

```
# statystyki sieci co 5 minut do pliku stat_net.png
```

```
*/1 * * * 3-5 $HOME/skrypt4.sh
```

Sprawdzenie, ściągamy plik ze stacji windows programem winscp:

Zainstalować paczkę openssh-server



Otworzyć plik w eksploratorze windows:

enp0s3 / 5 minute					2023-01-12 14:55	
time	rx	tx	total	avg. rate	■ rx	■ tx
2023-01-12						
11:55	253.73 KiB	20.12 KiB	273.85 KiB	7.48 kbit/s		
12:00	35.05 KiB	8.75 KiB	43.81 KiB	1.20 kbit/s		
12:05	44.86 MiB	98.84 KiB	44.96 MiB	1.26 Mbit/s		
13:20	7.40 KiB	8.23 KiB	15.63 KiB	426 bit/s		
13:25	35.50 KiB	12.33 KiB	47.83 KiB	1.30 kbit/s		
13:30	42.98 KiB	11.21 KiB	54.19 KiB	1.48 kbit/s		
13:35	35.28 KiB	8.96 KiB	44.24 KiB	1.21 kbit/s		

- f) uruchomienie skryptu arch_mysql.sh o zadanej godzinie, każdego 28 i 14 dnia miesiąca oraz w każdą środę, który wykona kopię zapasową bazy mysql w katalogu /tmp.

```
andrzej@servubu:~$ sudo apt install mysql-server -y
Reading package lists... Done
```

Test polecenia backup:

```
andrzej@servubu:~$ sudo mysqldump -u root --all-databases > /tmp/backup_mysql.sql
andrzej@servubu:~$ ls -al /tmp | grep mysql
-rw-r--r-- 1 andrzej sudo 1278965 Jan 13 08:29 backup_mysql.sql
```

Skrypt:

```
Open arch_mysql.sh
1 #!/bin/bash
2
3 sudo mysqldump -u root --all-databases > /tmp/backup_mysql.sql
```

Test działania skryptu:

```
andrzej@servubu:~$ rm /tmp/backup_mysql.sql
andrzej@servubu:~$ chmod u+x arch_mysql.sh
andrzej@servubu:~$ ./arch_mysql.sh
andrzej@servubu:~$ ls -al /tmp | grep mysql
-rw-r--r-- 1 andrzej sudo 1278965 Jan 13 08:33 backup_mysql.sql
```

Zadanie crona:

```
# kopia baz danych MySQL
*/1 * 13,14,28 * * $HOME/arch_mysql.sh
```

Sprawdzenie wykonania:

```
andrzej@servubu:~$ date
Fri 13 Jan 2023 08:39:10 AM CET
andrzej@servubu:~$ ls -al /tmp | grep mysql
-rw-r--r-- 1 andrzej sudo 1278965 Jan 13 08:39 backup_mysql.sql
```

- g) uruchomienie skryptu restart.sh przy każdym restarcie serwera, który wyśle maila z powiadomieniem
- h) wyślij mail do siebie każdego dnia 5 minut po północy.

10. Sprawdź zadania crontab wszystkich użytkowników

```
andrzej@servubu:~$ sudo ls -al /var/spool/cron/crontabs/  
total 12  
drwx-wx--T 2 root    crontab 4096 Jan  4 15:54 .  
drwxr-xr-x 5 root    root    4096 Feb 23  2022 ..  
-rw----- 1 andrzej crontab  350 Jan  4 15:54 andrzej
```

11. Z pomocą polecenia at zaplanuj zadania wykonane wcześniej dla crona.

12. Utwórz plik testat za 1 minutę:

```
andrzej@servubu:~$ at now + 1 minutes  
warning: commands will be executed using /bin/sh  
at> touch testat  
at> <EOT>  
job 8 at Fri Jan 20 10:14:00 2023  
andrzej@servubu:~$ at -l  
8          Fri Jan 20 10:14:00 2023 a andrzej  
andrzej@servubu:~$ atq  
8          Fri Jan 20 10:14:00 2023 a andrzej  
andrzej@servubu:~$ ls -al | grep testat  
-rw-r--r--  1 andrzej sudo          0 Jan 20 10:14 testat
```

13. Utwórz skrypt o nazwie arch.sh, który kopiuje zawartość katalogu ~/at_crontab/dane do katalogu ~/at_crontab/kopie.

14. Sprawdź zawartość katalogu /var/spool/cron/atjobs/

```
andrzej@servubu:~$ sudo ls -al /var/spool/cron/atjobs/  
total 20  
drwxrwx--T 2 daemon  daemon 4096 Jan 25 14:21 .  
drwxr-xr-x 5 root    root    4096 Feb 23  2022 ..  
-rw----- 1 andrzej daemon 4172 Jan 25 14:21 a0000a01c1eca5  
-rw----- 1 daemon  daemon   6 Jan 25 14:21 .SEQ
```

15. Utwórz skrypt o nazwie zmiana.sh, który zmieni nazwę pliku test2.txt na nowy.txt, przetestuj jego działanie, a następnie zaplanuj uruchomienie 22 i 23 grudnia, 5 minut po godzinie 12

16. Zaplanuj uruchomienie skryptu kom1.sh tylko w lutym, marcu, kwietniu i wrześniu o godzinie 11:30, ale tylko we wtorki i czwartki.

17. Z pomocą polecenia at zaplanuj:

- a) utwórz skrypt o nazwie zajetosc_dysku.sh, który tworzy plik na dysku z raportem o zajętości miejsca na dysku. Przetestuj jego działanie, a następnie zaplanuj jego uruchomienie podając godzinę, minuty i dzień tygodnia. Wykorzystaj narzędzie duf.

```
andrzej@servubu:~$ git clone https://github.com/muesli/duf.git
Cloning into 'duf'...
```

```
andrzej@servubu:~/duf$ sudo snap install go --classic
go 1.19.5 from Michael Hudson-Doyle (mwhudson) installed
```

```
andrzej@servubu:~/duf$ go build
go: downloading github.com/IGLOU-EU/go-wildcard v1.0.3
```

```
andrzej@servubu:~/duf$ ./duf
```

1 local device						
MOUNTED ON	SIZE	USED	AVAIL	USE%	TYPE	FILESYSTEM
/	20.6G	9.2G	10.4G	[#####.....] 44.7%	ext4	/dev/sda2

8 special devices						
MOUNTED ON	SIZE	USED	AVAIL	USE%	TYPE	FILESYSTEM
/dev	1.4G	0B	1.4G	[.....]	0.0%	devtmpfs
/dev/shm	1.4G	16.0K	1.4G	[.....]	0.0%	tmpfs
/run	285.4M	2.9M	282.5M	[.....]	1.0%	tmpfs

Skrypt:

```
andrzej@servubu:~/duf$ cat zajetosc_dysku.sh
/home/andrzej/duf/./duf > raport_dyskandrzej@
```

Uruchomienie i sprawdzenie:

```
andrzej@servubu:~/duf$ echo "./zajetosc_dysku.sh" | at now + 1 minutes
warning: commands will be executed using /bin/sh
job 11 at Wed Jan 25 15:14:00 2023
andrzej@servubu:~/duf$ at -l
10      Fri Jan 23 14:25:00 2026 a andrzej
11      Wed Jan 25 15:14:00 2023 a andrzej
andrzej@servubu:~/duf$ ls -al | grep raport
-rw-r--r--  1 andrzej sudo    3251 Jan 25 15:12  raport_dysk
andrzej@servubu:~/duf$ ls -al | grep raport
-rw-r--r--  1 andrzej sudo    3251 Jan 25 15:14  raport_dysk
```


18. Sprawdź jakie prawa posiadają pliki at, crontab.
19. Zezwól na używanie polecenia at sobie, a zabroń używania go użytkownikowi Blazej.

```
andrzej@servubu:~$ sudo cat /etc/at.deny
Blazej
alias
backup
bin
```

20. Zezwól na używanie polecenia crontab sobie, a zabroń używania go użytkownikowi blazej.

```
andrzej@servubu:/etc$ cat cron.deny
blazej
```

```
andrzej@servubu:/etc$ su - blazej
Password:
blazej@servubu:~$ crontab -e
You (blazej) are not allowed to use this program (crontab)
See crontab(1) for more information
```

21. Dodatkowe zadania:

- a) Wykonaj kopię zapasową bazy postgresql z pomocą crontab

Instalacja: apt install postgresql -y

Ustawienie hasła:

```
andrzej@servubu:~$ sudo -u postgres psql postgres
psql (12.13 (Ubuntu 12.13-0ubuntu0.20.04.1))
Type "help" for help.
```

```
postgres=# ALTER USER postgres PASSWORD 'postgres';
ALTER ROLE
postgres=# \q
```

Skrypt:

```
/home/andrzej/arch_postgresql.sh [----] 43 L:[ 1+ 1 2/ 3] *(76 / 112b) 0
PGPASSFILE=/home/andrzej/.pgpass
pg_dumpall -U postgres -h localhost -p 5432 -> $HOME/all_databases_postgres.sql
```

Utworzenie pliku .pgpass:

```
andrzej@servubu:~$ cat .pgpass
localhost:5432:*:postgres:postgres
```

Crontab:

```
# kopia baz danych PostgreSQL
*/1 * 25 * * $HOME/arch_postgresql.sh
```

Sprawdzenie działania:

```
andrzej@servubu:~$ ls -al | grep s_postgres
-rw-r--r--  1 andrzej sudo      1629 Jan 25 13:44 all_databases_postgres.sql
```

- b) Wykonaj kopię zapasową bazy MongoDB z pomocą at
- c) przetestuj polecenie at w Windows Server.
- d) inne podane przez nauczyciela

22. *sudo poweroff* (na koniec zajęć)