

Konta użytkowników

Przygotował Michał Tracewicz 2019

Spis treści

1. [Historia](#)
 2. [Pliki](#)
 3. [Administracja kontami użytkowników](#)
 4. [Zasoby systemowe](#)
 5. [Quoty](#)
 6. [Bibliografia](#)
-

Historia

System GNU-Linux powstał w roku 1991. Jest on oparty na systemie UNIX (lata 70-te dwudziestego wieku) wywodzącym się z Bell Labs. Co za tym idzie był on od początku projektowany z założeniem, że będzie to system przeznaczony na którym będzie możliwość pracy wielu użytkowników.

Pliki

Uprawnienia do plików

W systemach Linux możemy wyświetlić listę plików za pomocą polecenia ls.

```
ls -la
drwxr-xr-x 1 mtracewicz mtracewicz 4096 Oct  4 09:05 .oh-my-zsh
```

Kolejno od lewej wpis zawiera:

- Typ pliku:
 1. - dla plików zwykłych
 2. **d** dla katalogów
 3. **c** dla plików specjalnych
 4. **b** dla plików specjalnych przypisanych
 5. **l** dla łączy symbolicznych
- **Uprawnienia kolejno dla:**
 1. Użytkownika
 2. Grupy
 3. Innych

Dla każdej z tych kategorii możemy wyróżnić trzy rodzaje uprawnień

(Myślnik '-' oznacza, że dany użytkownik nie posiada danego prawa)

W wypadku gdy jest to plik nie będący katalogiem

- r - oznaczające możliwość czytania
- w - oznaczające możliwość edycji
- x - oznaczające możliwość uruchomienia

W wypadku przeciwnym

- r - oznaczające możliwość czytania plików zawartych w katalogu
- w - oznaczające możliwość tworzenia i usuwania plików w katalogu
- x - oznaczające możliwość dostępu do katalogu

Możemy to interpretować jako:

- r-x prawo dostępu do katalogu
- x prawo dostępu do plików o znanej nazwie

Uprawnienia te możemy również zapisać w postaci trzech liczby w systemie ósemkowym.

Gdzie:

0	---	4	r--
1	--x	5	r-x
2	-w-	6	rw-
3	-wx	7	rwX

- Liczba łączy
- Właściciel
- Grupa
- Objętość
- Data i godzina ostatniej modyfikacji
- Nazwa pliku

Możemy modyfikować uprawnienia dostępu za pomocą polecenia chmod.

Poniżej przykład użycia:

```
#nadajemy użytkownikowi możliwość uruchomienia pliku
chmod u+x exampleFile
#nadajemy grupie prawo edycji pliku
chmod g+w exampleFile
#odbieramy pozostałym użytkownikom możliwość czytania pliku
chmod o-r exampleFile
#odbieramy wszystkim użytkownikom możliwość uruchomienia pliku
chmod a-x exampleFile
#ustawiamy uprawnienia w formacie rwxr-xr-x
chmod 755 exampleFile
```

Mamy możliwość zmiany właściciela pliku oraz grupy za pomocą polecenia chown.

```
#zmieniamy właściciela pliku exampleFile na użytkownika mtracewicz a grupę na student.  
chown mtracewicz:student exampleFile  
#zmieniamy właściciela folderu exampleDir oraz wszystkich zawartych w nim plików na mtracewicz.  
chown -R mtracewicz exampleDir
```

Alternatywnie możemy zmienić grupę pliku za pomocą polecenia chgrp.

```
#zmieniamy grupę pliku example file na student  
chgrp student exampleFile
```

W systemie Linux informacje o użytkownikach znajdują się w plikach:

- /etc/passwd
- /etc/group
- /etc/shadow

Plik /etc/passwd

W tym pliku przechowywane są informacje o użytkownikach.

```
#Wszyscy użytkownicy mają możliwość odczytu pliku, gdybyśmy ją odebrali  
niebylibyśmy w stanie zmienić użytkownika a wiele aplikacji przestało by działać  
poprawnie nie mając dostępu do danych w nim dostępnych(stąd późniejszy podział  
na /etc/passwd i /etc/shadow)  
-rw-r--r-- 1 root root 1594 10-02 21:50 /etc/passwd  
#Przykładowy wpis w pliku /etc/passwd na Manjaro Linux  
mtracewicz:x:1000:1001:Michał Tracewicz:/home/mtracewicz:/bin/bash  
#|---1---|2|-3--|-4--|-----5-----|-----6-----|----7-----  
#Składnia:  
#1 - nazwa użytkownika  
#2 - hasło(zwykle znajdziemy tu x ponieważ aktualnie przechowuje się je w pliku  
/etc/shadow)  
#3 - id użytkownika  
#4 - id grupy  
#5 - komentarz/opis/informacja o użytkowniku  
#6 - folder domowy  
#7 - powłoka domyślna
```

Plik /etc/group

W tym pliku przechowywane są informacje o poszczególnych grupach w systemie. Dla przykładu

```
-rw-r--r-- 1 root root 988 10-03 14:42 /etc/group
#Przykładowy wpis w pliku /etc/group na Manjaro Linux
sys:x:3:bin,mtracewicz
#|1|2|3|-----4-----
#Składnia
#1 - nazwa grupy
#2 - hasło(zwykle puste ale może zawierać zaszyfrowane hasło)
#3 - id grupy
#4 - lista użytkowników należących do grupy
```

Możemy sprawdzić do jakich grup należy dany użytkownik poprzez użycie polecenia groups.

```
#Przykład użycia polecenia groups dla użytkownika mtracewicz
groups mtracewicz
wheel lp sys network power autologin vboxusers mtracewicz
```

Plik /etc/shadow

W tym pliku przechowujemy hasła użytkowników.

```
#Możemy zauważyć, że w przeciwieństwie do poprzednich plików plik /etc/shadow
może być zarówno czytany jak i edytowany przez użytkownika root
-rw----- 1 root root 922 10-02 21:50 /etc/shadow
#Przykładowy wpis w pliku(wzięty z https://www.slashroot.in/how-are-passwords-
stored-linux-understanding-hashing-shadow-utils i delikatnie zmodyfikowany)
testUser:$1$Etg2ExUZ$F9NTP7omafhKI1qABMqng1:15651:0:99999:7:::
#-1-|-----2-----|--3--|4|--5--|6|7|8|9
#1 - nazwa użytkownika
#2 - zaszyfrowane hasło(poniżej przykładu znajduje się informacja o tym jak
wygląda ten proces)
#3 - ile dni minęło od ostatniej zmiany hasła
#4 - ile minimalnie dni jest wymaganych między zmianami hasła(jak często można
zmieniać hasło)
#5 - ile maksymalnie dni jest dopuszczalne między zmianami hasła
#6 - na ile dni przed następną wymaganą zmianą hasła użytkownik dostanie
ostrzeżenie
#7 - ile dni po wygaśnięciu hasła konto będzie wyłączone
#8 - po ilu dniach od 01.01.1970r. konto zostanie wyłączone
#9 - pole jeszcze nie obecnie używane
```

W jaki sposób hasła są zabezpieczane?

Hasło przechowywane w pliku /etc/shadow możemy podzielić na trzy części rozdzielone znakiem '\$'. Przyjmuje ono postać \$ID\$SALT\$HASHED.

Algorytm hashujący - algorytm który z podanych danych tworzy unikatowy ciąg znaków zadanej długości. Jest to funkcja, której nie da się w prosty sposób odwrócić tzn. znając hash nie możemy odzyskać danych wejściowych (To odróżnia algorytm hashujący od szyfrującego, ten drugi jest łatwo odwracalny gdy znamy odpowiedni algorytm deszyfrujący).

ID jest to wartość wskazująca jakiego algorytmu hashującego użyto. Może on przyjąć wartości:

- 1 - oznacza algorytm MD5(Nie jest zalecane jego użycie, obecnie jest łatwy do złamania)

- 2 - oznacza algorytm Blowfish
- 2a - oznacza algorytm eksblowfish
- 5 - oznacza algorytm SHA-256
- 6 - oznacza algorytm SHA-512

Salt jest to losowo wygenerowany ciąg znaków, który jest łączony z hasłem użytkownika w celu zwiększenia bezpieczeństwa.

HASHED jest to wartość wynikowa algorytmu hashującego na hasło użytkownika połączonym z saltem.

Co daje nam salt?

Salt pomaga nam zabezpieczyć nasze hasła przed atakami typu dictionary attack czy rainbow table (więcej o tym w następnym podpunkcie). Dzięki zastosowaniu wartości salt nawet dwa dokładnie te same hasła będą posiadały inny hash. Co za tym idzie nawet jeżeli osobie atakującej udało się złamać jedno hasło nie będzie ona w stanie znaleźć osoby o identycznym hasle ponieważ ich zahaszowana wartość będzie inna.

Jak można łamać hasła?

Najprostszym sposobem łamania haseł są tak zwane dictionary attack i rainbow table.

Pierwszy z nich to atak oparty na prostej metodzie siłowej gdzie znając algorytm hashujący próbujemy użyć go na wszystkich prawdopodobnych hasłach (najczęściej robi się to sprawdzając listę najczęstszych haseł oraz dodając do niej te same hasła tylko ze zmienioną wielkością liter czy podmieniając litery na cyfry np. 'A' -> 4, 'O' -> 0 itp.) i znaleźć takie, które zgadza się z jednym z tych które pozyskaliśmy.

Drugi sposób to pozyskanie bazy w której najpopularniejsze hasła są już zahaszowane wraz z informacją tym jaki algorytm został użyty. Następnie sprawdzamy czy, któryś z posiadanych przez nas hashy znajduje się w tej bazie i odczytujemy z niej hasło.

W pierwszym przypadku zużywamy niewiele pamięci jednak bardzo dużo mocy obliczeniowej, w drugim ataku jest dokładnie odwrotnie. Przed oboma tymi atakami pomaga nam bronić się wartość salt. Dzięki generowaniu losowej wartości do naszych haseł mamy niemal pewność, że hash, który uzyskamy (nawet jeżeli użytkownik ustawi sobie hasło = hasło123!) nie znajdzie się w żadnej z rainbow tables. W przypadku dictionary attack dodanie wartości salt masowo zwiększa ilość możliwości, które atakujący musi sprawdzić a co za tym idzie zwiększamy czas, który musi poświęcić na próbę złamania każdego z haseł.

Czym jest silne hasło?

Silne hasło to takie które zawiera minimum osiem znaków, zarówno wielkie jak i małe litery, znaki specjalne i cyfry.

Jeżeli nasze hasło zawiera tylko 8 małych liter to jest ich możliwie 26^8 , natomiast w wypadku bezpiecznego hasła jest ich minimum 56^8 (liczba ta jest większa zależnie od tego jakie znaki dopuszczamy jako znaki specjalne).

Dodatkowo należy pamiętać, że długość hasła ma istotny wpływ na jego bezpieczeństwo. Jak już pokazaliśmy ośmioznakowych haseł jest $\sim 56^8$ natomiast dodanie np. czterech znaków znacząco zwiększa ilość możliwości 56^{12} . Pokazuje to, że każdy kolejny znak zwiększa ilość obliczeń, którą musi wykonać ktoś, kto próbuje zgadnąć nasze hasło.

Warto także pamiętać o tym, że hasło nie powinno zawierać żadnych danych z nami związanych takich jak imię, nazwisko czy rok urodzenia.

Administracja kontami użytkowników

Wyświetlanie listy aktywnych użytkowników

W systemie Linux możemy wyświetlić listę aktywnych użytkowników za pomocą polecenia `users`.

```
users
#W normalnym systemie wynikiem tego polecenia jest lista aktualnie zalogowanych
użytkowników
test testUser exampleUser
```

Polecenie to nie zawiera żadnych opcji.

Wyświetlanie ostatnich logowań użytkowników

W systemie Linux możemy wyświetlić listę ostatnich logowań użytkowników za pomocą polecenia `last`.

```
#polecenie wyświetli logowania użytkownika mtraciewicz w kolejności od
najstarszych do najnowszych możemy także wyświetlić dla konkretnego tty/host
last mtraciewicz
#przykładowy wpis
mtraciewicz pts/9      188.147.44.127.nat.umn.pl  4 paź 09:15 - 09:18  (00:02)
#----1----|--2-----|-----3-----|-----4-----|
#1 - nazwa użytkownika
#2 - tty(nazwa terminalu)
#3 - host z którego użytkownik się loguje/miejsce dostępu
#4 - data początku - końca logowania i w nawiasie czas trwania
```

Dodawanie użytkowników

W systemie Linux możemy dodać użytkownika za pomocą polecenia `useradd`.

```
#polecenie, które doda do systemu użytkownika test, pobierze domyślne wartości z
pliku /etc/default/useradd może zostać wykonane tylko przez użytkownika root lub
użytkownik posiadający uprawnienia do polecenia sudo
useradd test
#jeżeli chcemy utworzyć katalog domowy użytkownikowi musimy użyć opcji -m
useradd -m test
#jeżeli użyjemy opcji -d możemy utworzyć katalog domowy w miejscu innym niż
domyślne
#jeżeli chcemy dodać użytkownika do grup użyjemy opcji -G
useradd test -G student,inf
#w tym wypadku utworzymy użytkownika test i dodamy go do grup student i inf
#jeżeli chcemy ustawić np. po ilu dniach wygasa hasło użyjemy opcji -K
useradd test -K PASS_MAX_DAYS = 3
#jeżeli chcemy dodać komentarz jak np. imię i nazwisko to użyjemy opcji -c
useradd test -c "Jan Kowalski"
```

Usuwanie użytkowników

W systemie Linux możemy usunąć użytkownika za pomocą polecenia `userdel`

```
#tym poleceniem usuniemy użytkownika test, może zostać wywołane tylko przez
użytkownika root lub użytkownik posiadający uprawnienia do polecenia sudo
userdel test
#jeżeli chcemy usunąć także katalog domowy użytkownika użyjemy opcji -r
userdel -r test
```

Modyfikacja użytkowników

W systemie Linux możemy modyfikować użytkownika za pomocą polecenia usermod.

```
#tym poleceniem zmienimy katalog domowy użytkownika test na katalog /test
usermod -d /test test
#jeżeli chcemy wraz ze zmianą katalogu domowego przenieść do niego pliki ze
starego używamy opcji -m
usermod -d /test -m test
#tym poleceniem zmienimy login użytkownika test na jankowalski
usermod -l test jankowalski
#tym poleceniem zmienimy id użytkownika test na 1000
usermod -u 1000 test
#tym poleceniem zmienimy główną grupę użytkownika test na pracownik(grupa musi
już istnieć)
usermod -g pracownik test
#tym poleceniem dodamy wiele grup(student,informatyka) dla użytkownika test.
Opcja -a sprawia, że użytkownik nie utraci obecnie przypisanych grup
usermod -a -G student,informatyka test
#tym poleceniem zmienimy datę wygaśnięcia konta użytkownika test na pierwszy
stycznia 2020. Data musi być w formacie YYYY-MM-DD
usermod -e 2020-01-01 test
#tym poleceniem zmienimy powłokę użytkownika test na zsh
usermod -s /bin/zsh test
```

Zmiany hasła

W systemie Linux możemy modyfikować hasło użytkownika za pomocą polecenia passwd.

```
#Każdy użytkownik może zmienić własne hasło
passwd
#Wyświetli nam się taki komunikat
Changing password for mtracewicz.
#Zostaniemy poproszeni o aktualne hasło
Current password:
#Następnie o nowe hasło
New password:
#Oraz powtórzenie w celu potwierdzenia
Retype new password:
#Użytkownik root może zmodyfikować hasło dowolnego użytkownika. Tym poleceniem
zmienimy hasło użytkownika test(jako root nie zostaniemy zapytani o poprzednie
hasło)
passwd test
#Polecenie passwd pozwala nam też usunąć hasło opcją -d
passwd -d test
```

Jak wymusić zmianę hasła?

Aby wymusić zmianę hasła możemy użyć wcześniej wspomnianego polecenia `passwd` lub dedykowanego polecenia `chage`.

```
# Aby wymusić zmianę hasła przy pierwszym logowaniu hasłem nadanym przez root-a
możemy użyć opcji -e
passwd -e test
# Polecenie change służy do zarządzania wygasaniem haseł. Możemy użyć polecenia
change do wyświetlenia aktualnych informacji o danych związanych z hasłem
użytkownika w ten sposób:
chage -l mtracewicz
# Możemy zmienić maksymalną ilość dni między zmianami hasła z opcją -M. W tym
przykładzie ustawimy, że użytkownik mtracewicz musi zmienić hasło co maksymalnie
5 dni
chage -M 5 mtracewicz
# Jeżeli nie chcemy aby użytkownik zmieniał hasło codziennie możemy użyć opcji -
m. W tym przykładzie zmienimy, że użytkownik mtracewicz będzie mógł zmienić hasło
najczęściej co dwa dni.
chage -m 2 mtracewicz
```

Blokowanie / odblokowanie konta

Wcześniej wymienionym poleceniem `usermod` możemy zablokować lub odblokować użytkownika.

```
# tym poleceniem blokujemy użytkownika
usermod -L test
# tym poleceniem odblokujemy użytkownika
usermod -U test
```

Zmiana tożsamości użytkownika

W systemie Linux mamy dwa polecenia służące do zmiany tożsamości: `sudo`, `su`.

Różnica między nimi polega na tym, że polecenie `sudo` służy do wykonania polecenia jako inny użytkownik zaś polecenie `su` służy do zmiany użytkownika

Polecenie su:

```
# Wykonanie polecenia su bez argumentów zmini użytkownika na root
su
# Możemy dopisać nazwę użytkownika aby wybrać na jakiego użytkownika chcemy
zminić
su testUser
# Użyjemy opcji -s kiedy chcemy wybrać powłokę
su -s /bin/zsh
# Opcja -s wybierze powłokę w kolejności:
# 1. wprowadzona przez nas w poleceniu
# 2. ze zmiennej $SHELL (jeżeli użyto opcji --preserve-environment, opcja ta
zachowuje nasze zmienne środowiskowe z wyjątkiem $PATH i $IFS)
# 3. odczytaną z pliku /etc/passwd
```

Polecenie `su` możemy konfigurować za pomocą pliku `/etc/login.defs`. Możemy tam np. ustawić logowanie do pliku wszystkich poleceń wykonanych przez użytkownika po użyciu polecenia `su`.

Polecenie sudo:


```
#W tym przykładzie użyjemy polecenia sudo aby zainstalować dodatkowe
oprogramowanie
sudo dnf install vim
#Możemy użyć opcji -u aby wybrać jako jaki użytkownik chcemy wykonać dane
polecenie
sudo -u test vim test.c
#Możemy użyć opcji -g aby wykonać polecenie jakbyśmy byli członkami innej grupy
sudo -g 999 vim test.c
```

Polecenie sudo jest konfigurowane w pliku /etc/sudoers. Plik jest podzielony na trzy sekcje: defaults, aliases oraz user specifications. Sekcja defaults zawiera konfiguracje, które będą automatycznie dopisywane do każdego rekordu, mogą one jednak być nadpisywane dla konkretnego wpisu. Sekcja aliases zawiera zmienne, które służą do grupowania wielu nazw do jednego słowa. Istnieją cztery typy aliasów:

- User_Alias - łączymy kilku użytkowników w grupę np.: User_Alias testowi = test1, test2. Nie musimy tu redefiniować grup, które zdefiniować w systemie. Aby użyć grupy systemowej wstawimy przed jej nazwą '%' np.: User_Alias testowi = %testowi.
- Runas_Alias - jak wyżej z różnicą, że jest to grupa użytkowników jako, którzy polecenie ma być wykonane.
- Host_Alias - służy do grupowania hostów z, których użytkownik wykonujący polecenie sudo się loguje.
- Cmnd_Alias - służy do grupowania poleceń np.: Cmnd_Alias fileList = /bin/ls

Dla każdego z tych typów aliasów istnieje wbudowany alias ALL. Dodatkowo dodanie '!' przed nazwą polecenia oznacza, że użytkownik nie będzie mógł go wykonać

W sekcji user specifications zawieramy konkretne wpisy opisujące możliwości danego użytkownika.

```
#Wpis ma postać
user host = (runas) command[, command, ...]
#Przykładowy wpis
testUser ALL = (%students) /bin/ls
#----1--|-2--|-----3-----|---4---|
#1. użytkownik/grupa systemowa(poprzedona %)/User_Alias, któremu przyznajemy
prawa wykonania sudo(w tym wypadku testUser)
#2. host/Host_Alias z, którego może on wykonać to polecenie(w tym wypadku
dowolny)
#3. może wykonać jako użytkownik/grupa systemowa(poprzedona %)/User_Alias(w tym
wypadku grupa students)
#4. polecenia do których otrzymuje dostęp (w tym wypadku polecenie ls)
testUser2 ALL = (ALL) ALL,!/bin/vim
#W powyższym przykładzie daliśmy prawo wykonania wszystkich poleceń z wyjątkiem
polecenia vim użytkownikowi testUser2 na wszystkich hostach jako dowolny
użytkownik
```

Warto zaznaczyć, że domyślnie polecenie sudo pyta użytkownika o jego hasło, po czym zapamiętuje to hasło na pięć minut.

Dobre praktyki

Przy konfiguracji pliku `/etc/sudoers` warto pamiętać o kilku prostych zasadach aby polepszyć bezpieczeństwo naszego systemu. Przede wszystkim warto wyłączyć każdemu z użytkowników możliwość użycia polecenia `su` przez polecenie `sudo`. Jest to ważne ponieważ w przeciwnym wypadku dowolny użytkownik może się zalogować jako `root` używając swojego hasła.

```
#W wypadku braku tego zabezpieczenia poniższym poleceniem możemy się zalogować
na użytkownika root z użyciem hasła do naszego konta!
sudo su
```

Dodatkowo warto wyłączyć możliwość uruchamiania plików z katalogów do których zwykły użytkownik ma prawo zapisu. Dzięki temu zwykły użytkownik nie będzie w stanie uruchomić programów pobranych z Internetu. Możemy to zrobić poprzez dodanie aliasu `"Cmd_Alias NAZWA_ALIASU = /home/, /tmp/, /var/tmp/*"` oraz dodając przeciwny alias do zaufanych lokacji programów `"Cmd_Alias BEZPIECZNE = /sbin:/bin:/usr/sbin:/usr/bin"`;

```
#Przykładowy plik /etc/sudoers (wzorowany na
https://stelfox.net/blog/2016/02/better-practices-with-sudo/)
# /etc/sudoers
#Alias do poleceń, których nie chcemy aby użytkownicy używali, w naszym wypadku
jest to polecenie su z wyżej wymienionego powodu
Cmd_Alias BLACKLISTED_APPS = /bin/su
#Alias do folderów z, których nie chcemy aby użytkownik mógł uruchamiać programy
Cmd_Alias USER_WRITEABLE = /home/*, /tmp/*, /var/tmp/*
#Dopisujemy do wszystkich rekordów foldery z, których chcemy pozwolić uruchamiać
programy
Defaults secure_path = /sbin:/bin:/usr/sbin:/usr/bin
#Pozwalamy użytkownikowi root robić wszystko
root ALL = (ALL) ALL
#Aplikujemy nasze zasady dla wszystkich pozostałych użytkowników
%zwykliUzytkownicy ALL = (root) ALL, !BLACKLISTED_APPS, !USER_WRITABLE
```

Dodawanie grup

W systemie Linux możemy dodać grupę za pomocą polecenia `useradd`.

```
#Tym poleceniem dodamy grupę testGroup
groupadd testGroup
#Z opcją -g możemy sami wybrać id grupy(musi być unikatowe i nie ujemne)
groupadd -g 999 testGroup
```

Usuwanie grup

W systemie Linux możemy usunąć grupę za pomocą polecenia `groupdel`.

```
#Tym poleceniem usuniemy grupę testGroup, grupa musi istnieć i my jako
administratorzy musimy zadbać aby grupa, która usuwamy nie była główną grupą
dla żadnego z użytkowników
groupdel testGroup
```

Modyfikacja grup

W systemie Linux możemy zmodyfikować grupę za pomocą `groupmod`.

```
#Możemy zmodyfikować id grupy przy użyciu opcji -g
groupmod -g 999 testGroup
#Możemy też zmodyfikować nazwę grupy za pomocą opcji -n. W tym przykładzie
zminimy nazwę grupy testGroup na myGroup
groupmod -n myGroup testGroup
```

Zasoby systemowe

Listowanie procesów i ich zasobów

W systemie Linux istnieje kilka możliwości wyświetlenia aktywnych procesów. Możemy użyć do tego polecenia ps, top i fuser.

PS

```
#Podstawowe wywołanie
ps
#Wyjście polecenia ma format i wyświetla tylko procesy aktualnego użytkownika
#PID TTY          TIME CMD
#Aby wyświetlić wszystkie procesy w systemie możemy użyć dwóch wersji polecenia
ps (posiada ono różne wersje w standardzie UNIX, BSD i GNU)
#Różnica między tymi poleceniami jest format wyświetlonego wyjścia
ps -ely
#Wyjście polecenia ma format
#S  UID  PID  PPID  C PRI  NI   RSS   SZ  WCHAN  TTY          TIME CMD
ps -axu
#Wyjście polecenia ma format
#USER      PID %CPU %MEM    VSZ   RSS TTY      STAT START   TIME COMMAND
#Jeżeli chcemy wyświetlić procesy danego użytkownika używamy opcji -U
ps -U testUser
```

TOP

```
#Polecenie top w przeciwieństwie do polecenia ps jest dynamicznie aktualizowane
i wyświetla aktualny stan zasobów systemu
top
#Przykładowy wynik polecenia top, widzimy tu status aktualnie uruchomionych
zadań, obciążenie CPU, pamięć wolną, zajęta a także przeniesioną do swap
top - 14:20:31 up 1:41, 0 users, load average: 0.52, 0.58, 0.59
Tasks: 5 total, 1 running, 4 sleeping, 0 stopped,
0 zombie          %Cpu(s): 1.8 us, 2.3 sy, 0.0 ni,
95.1 id, 0.0 wa, 0.8 hi, 0.0 si, 0.0 st          KiB Mem :
8241956 total, 4504524 free, 3508080 used, 229352 buff/cache
KiB Swap: 25165824 total, 25064552 free, 101272 used. 4600144
avail Mem

      PID USER      PR  NI
VIRT  RES  SHR S  %CPU %MEM    TIME+  COMMAND
root    20   0   8892  296  260 s   0.0  0.0   0:00.04 init
        63 root    20   0   19464  744  576 s   0.0  0.0
0:00.00 sshd
        160 s   0.0  0.0   0:00.01 init
        20   0   17012 3696 3588 s   0.0  0.0   0:00.33 bash
        442 mtracew+ 20   0   17620 2032 1504 R   0.0  0.0   0:00.01 top
```

#W przeciwieństwie do poprzednich poleceń fuser nie wyświetla listy aktualnie działających procesów lecz to jakie procesy aktualnie korzystają z danego pliku (plik ten może być katalogiem, zwykłym plikiem, plikiem wykonywalnym, etc.) lub gniazda.

#Wywołanie bez opcji spowoduje pokazanie pomocy, aby program działał musimy wskazać plik

fuser .

#Powyższe polecenie wskazuje nam jakie procesy korzystają z obecnego katalogu. W wyniku otrzymamy listę pidów zakończonych literą wskazującą typ dostępu, może on przyjmować następujące wartości.

#1. c - obecny katalog

#2. e - plik wykonywalny jest uruchomiony

#3. f - otwarty plik (omijany w standardowym wyświetlaniu)

#4. F - plik otwarty do zapisu (omijany w standardowym wyświetlaniu)

#5. r - folder root

#6. m - zmapowany plik lub biblioteka współdzielona

#Przykładowe wyjście

```
/home/mtracewicz: 1490c 1491c 1493c 1496c 1528c 1535c 1601c 1631c 1641c
1646c 1647c 1652c 1655c 1659c 1660c 1662c 1665c 1681c 1685c 1689c
1695c 1704c 1708c 1718c 1723c 1728c 1733c 1738c 1745c 1746c 1748c
1752c 1754c 1756c 1759c 1760c 1764c 1766c 1772c 1783c 1790c 1796c
1799c 1805c 1807c 1836c 1874c 1886c 1934c 1975c 1982c 1983c 1987c
1991c 1999c 2015c 2110c 2132c 2151c
```

#Możemy użyć flagi -u pokaże nam użytkownika do, którego należy dany proces

```
/home/mtracewicz: 1490c(mtracewicz) 1491c(mtracewicz) 1493c(mtracewicz)
1496c(mtracewicz) 1528c(mtracewicz) 1535c(mtracewicz) 1601c(mtracewicz)
1631c(mtracewicz) 1641c(mtracewicz) 1646c(mtracewicz) 1647c(mtracewicz)
1652c(mtracewicz) 1655c(mtracewicz) 1659c(mtracewicz) 1660c(mtracewicz)
1662c(mtracewicz) 1665c(mtracewicz) 1681c(mtracewicz) 1685c(mtracewicz)
1689c(mtracewicz) 1695c(mtracewicz) 1704c(mtracewicz) 1708c(mtracewicz)
1718c(mtracewicz) 1723c(mtracewicz) 1728c(mtracewicz) 1733c(mtracewicz)
1738c(mtracewicz) 1745c(mtracewicz) 1746c(mtracewicz) 1748c(mtracewicz)
1752c(mtracewicz) 1754c(mtracewicz) 1756c(mtracewicz) 1759c(mtracewicz)
1760c(mtracewicz) 1764c(mtracewicz) 1766c(mtracewicz) 1772c(mtracewicz)
1783c(mtracewicz) 1790c(mtracewicz) 1796c(mtracewicz) 1799c(mtracewicz)
1805c(mtracewicz) 1807c(mtracewicz) 1836c(mtracewicz) 1874c(mtracewicz)
1886c(mtracewicz) 1934c(mtracewicz) 1982c(mtracewicz) 1983c(mtracewicz)
1987c(mtracewicz) 1991c(mtracewicz) 1999c(mtracewicz) 2110c(mtracewicz)
2132c(mtracewicz) 2151c(mtracewicz)
```

#Możemy także użyć opcji -v aby pokazać rozbudowane wyjście

	USER	PID	ACCESS	COMMAND
/home/mtracewicz:	mtracewicz	1490	..c..	dbus-broker-lau
	mtracewicz	1491	..c..	dbus-broker
	mtracewicz	1493	..c..	gdm-wayland-ses
	mtracewicz	1496	..c..	gnome-session-b
	mtracewicz	1528	..c..	gvfsd
	mtracewicz	1535	..c..	gvfsd-fuse
	mtracewicz	1601	..c..	gnome-shell
	mtracewicz	1631	..c..	Xwayland
	mtracewicz	1641	..c..	at-spi-bus-laun
	mtracewicz	1646	..c..	dbus-broker-lau
	mtracewicz	1647	..c..	dbus-broker
	mtracewicz	1652	..c..	at-spi2-registr
	mtracewicz	1655	..c..	ibus-daemon
	mtracewicz	1659	..c..	ibus-dconf

```

mtracewicz 1660 ... ibus-extension-
mtracewicz 1662 ... ibus-x11
mtracewicz 1665 ... ibus-portal
mtracewicz 1681 ... xdg-permission-
mtracewicz 1685 ... gnome-shell-cal
mtracewicz 1689 ... evolution-sourc
mtracewicz 1695 ... goa-daemon
mtracewicz 1704 ... dconf-service
mtracewicz 1708 ... gvfs-udisks2-vo
mtracewicz 1718 ... gvfs-goa-volume
mtracewicz 1723 ... goa-identity-se
mtracewicz 1728 ... gvfs-gphoto2-vo
mtracewicz 1733 ... gvfs-mtp-volume
mtracewicz 1738 ... gvfs-afc-volume
mtracewicz 1745 ... gsd-smartcard
mtracewicz 1746 ... gsd-keyboard
mtracewicz 1748 ... gsd-power
mtracewicz 1752 ... gsd-a11y-settin
mtracewicz 1754 ... gsd-sound
mtracewicz 1756 ... gsd-media-keys
mtracewicz 1759 ... gsd-print-notif
mtracewicz 1760 ... gsd-clipboard
mtracewicz 1764 ... gsd-wacom
mtracewicz 1766 ... gsd-mouse
mtracewicz 1772 ... gsd-rfkill
mtracewicz 1783 ... gsd-color
mtracewicz 1790 ... gsd-xsettings
mtracewicz 1796 ... gsd-screensaver
mtracewicz 1799 ... gsd-datetime
mtracewicz 1805 ... gsd-sharing
mtracewicz 1807 ... gsd-housekeepin
mtracewicz 1836 ... evolution-calen
mtracewicz 1874 ... evolution-addre
mtracewicz 1886 ... gsd-printer
mtracewicz 1934 ... ibus-engine-sim
mtracewicz 1982 ... abrt-applet
mtracewicz 1983 ... gsd-disk-utilit
mtracewicz 1987 ... gnome-software
mtracewicz 1991 ... evolution-alarm
mtracewicz 1999 ... tracker-miner-f
mtracewicz 2110 ... gvfsd-metadata
mtracewicz 2132 ... gnome-terminal-
mtracewicz 2151 ... zsh

```

Limitowanie zasobów systemowych dla użytkownika

W systemie Linux możemy użyć polecenia ulimit aby nakładać limit na zasoby systemowe.

```

#Możemy wyświetlić obecne limity dla zwykłego użytkownika użyjemy opcji -a
ulimit -a
#Przykładowe wyjście
-t: cpu time (seconds)          unlimited
-f: file size (blocks)         unlimited
-d: data seg size (kbytes)     unlimited
-s: stack size (kbytes)       8192
-c: core file size (blocks)    unlimited
-m: resident set size (kbytes) unlimited

```

```
-u: processes 19678
-n: file descriptors 1024
-l: locked-in-memory size (kbytes) 64
-v: address space (kbytes) unlimited
-x: file locks unlimited
-i: pending signals 19678
-q: bytes in POSIX msg queues 819200
-e: max nice 0
-r: max rt priority 0
-N 15: unlimited
```

#W tym wyjściu widzimy też możliwe dla nas opcje. Widzimy np. opcję "-u", która pozwala nam zmienić limit procesów. Jeżeli użyjemy którejś z tych opcji bez wpisania wartości wyświetli on aktualny miękki limit.

```
ulimit -u
19678
```

#W systemie rozróżniamy dwa typy limitów:

miękki - jest on pilnowany przez jądro systemu

twardy - służy on za górną wartość dla limitu miękkiego

#Teraz ustawiamy limit(nie podając opcji "S" lub "H" ustawimy naraz oba limity,miękki i twardy)

```
ulimit -u 50
```

#Opcja "S" pozwala ustawić limit miękki

```
ulimit -Su 50
```

#A "H" twardy(ważne jest aby "H"/"S" znajdowały się przed inną opcją jak "u" inaczej zamiast ustawić nowy limit wyświetlimy odpowiedni limit a wartość którą chcemy nadać zostanie zignorowana)

```
ulimit -uS 50
```

#Powyższe polecenie wyświetli miękki limit dla procesów dla aktualnego użytkownika

Jeżeli chcemy ustawić dla konkretnych użytkowników musimy to zrobić w pliku /etc/security/limits.conf.

```
#wpis w tym pliku ma format:
#<domain> <type> <item> <value>
#Gdzie domian to użytkownik/grupa
#Type to "soft"/"hard" (wyjaśnione w poprzednim przykładzie)
#Item to zasobób (lista zasobów jest widoczna w poprzednim przykładzie jako lista opcji)
#Value wskazuje na ile ustawiamy limit
#Przykładowy wpis
mtracewicz soft nproc 50
```

Quoty

Bibliografia

Polecenie last

- <https://www.golinuxhub.com/2014/05/how-to-check-last-login-time-for-users.html>
- man last

Polecenie users

- man users

SU

- [man su](#)

Sudo

- <https://www.lifewire.com/what-to-know-sudo-command-3576779>
- <https://www.ixsystems.com/blog/best-practices-in-unix-access-control-with-sudo/>
- <https://stelfox.net/blog/2016/02/better-practices-with-sudo/>

Dostęp do plików

- <http://www.penguintutor.com/linux/file-permissions-reference>
- <http://mediologia.pl/katalogi-i-pliki-linux/2-4-atrybuty-plikow-uzywanych-w-systemie-linux-polecenie-ls>
- <https://www.hostingadvice.com/how-to/change-file-ownershipgroups-linux/>

Pliki z informacjami o użytkownikach/grupach

- <https://www.cyberciti.biz/faq/understanding-etcgroup-file/>
- <http://www.yourownlinux.com/2015/07/etc-passwd-file-format-in-linux-explained.html>

Hasła użytkowników

- <https://www.cyberciti.biz/faq/understanding-etcshadow-file/>
- [https://www.slashroot.in/how-are-passwords-stored-linux-understanding-hashing-shadow-utils\](https://www.slashroot.in/how-are-passwords-stored-linux-understanding-hashing-shadow-utils)
- <https://blog.jscrambler.com/hashing-algorithms/>

Tworzenie, usuwanie i modyfikacja kont użytkowników

- <https://www.lifewire.com/create-users-useradd-command-3572157>
- <https://www.linuxnix.com/delete-user-account-linux/>
- <https://www.itzgeek.com/how-tos/linux/how-to-modify-user-accounts-in-linux-using-usermod-command.html>

Blokowanie użytkowników

- <https://www.linuxnix.com/lock-user-account-linux/>
- <https://www.2daygeek.com/lock-unlock-disable-enable-user-account-linux/>

Procesy

- <https://linux.101hacks.com/unix/fuser/>
- <https://linux.101hacks.com/unix/top/>
- <https://linux.101hacks.com/monitoring-performance/ps-command-examples/>

Zasoby

- <https://ss64.com/bash/ulimit.html>
- <https://www.networkworld.com/article/2693414/setting-limits-with-ulimit.html>

Quoty

- <https://www.linux.com/tutorials/step-step-using-user-quotas-linux/>
- <https://www.looklinux.com/how-to-manage-disk-quota-in-linux/>

- https://docs.fedoraproject.org/en-US/Fedora/14/html/Storage_Administration_Guide/ch-disk-quotas.html
- <https://www.howtoforge.com/tutorial/linux-quota-ubuntu-debian/>
- https://wiki.archlinux.org/index.php/Disk_quota
- <https://www.itworld.com/article/2811509/storage-quotas---hard-vs--soft---explained.html>
- https://en.wikipedia.org/wiki/Disk_quota#Common_Unix_disk_quota_utilities

Linux

- <https://en.wikipedia.org/wiki/Linux>
- [The Complete History of Linux \(Abridged\) -Bryan Lunduke](#)