

Notatki 1

Wstęp, podstawy Linuxa

Zasady zaliczenia, zakres i materiały:

https://brain.fuw.edu.pl/edu/index.php/%22Programowanie_z_Pythonem3%22

Dystrybucja Linuxa – system zbudowany na bazie jądra Linuxa oraz menadżera pakietów (czyli programu pozwalającego na instalację i zarządzanie programami, np. apt, dnf, pacman...) współpracującego z repozytorium oprogramowania. Istnieją setki dystrybucji: Debian, Ubuntu, Mint, Fedora, Arch, Gentoo, Kali itp. (<https://distrowatch.com/>), które różnią się między sobą przede wszystkim wyborem konkretnego menadżera pakietów i oprogramowania w repozytorium, a także pewnymi rozwiązaniami konfiguracji. W chwili obecnej na komputerach w pracowniach FUW zainstalowana jest Fedora 42 (a na niektórych innych dostępnych Państwu komputerach także Ubuntu).

Przydatne polecenia (*nawiasy ostrokatne <,> oznaczają poniżej argumenty polecenia i nie należy ich wpisywać przy wywoływaniu*):

- `man/info <nazwa_polecenia>` – dokumentacja
- `ls <katalog>` – wypisuje zawartość katalogu (bez podania opcji pokazuje bieżący, opcja `-l` prezentuje wyniki w postaci listy, `-h` dodatkowo wypisuje wielkości plików w bardziej przejrzystym dla człowieka formacie, `-R` wypisuje również zawartość podkatalogów)
- `pwd` – pokazuje gdzie się znajdujemy
- `cd <katalog>` – przechodzi do innego katalogu (w przypadku braku podania celu przenosi nas do katalogu domowego, bieżący katalog oznaczamy jako `.` i katalog wyżej jako `..`)
- `mkdir <katalog>` – tworzy nowy katalog
- `touch <plik>` – tworzy nowy (pusty) plik
- `rm <plik>` – usuwa plik (opcja `-r` pozwala też usunąć katalog, zaś `-f` pomija pytania o usunięcie oraz ignoruje nieistniejące pliki)
- `cp <plik_a> <plik_b>` – kopiuje plik a do pliku b (opcja `-r` pozwala skopiować katalog)
- `mv <plik_a> <plik_b>` – przenosi plik a do pliku b (również działa z katalogami)
- `top/htop` – pokazuje aktywne procesy
- `kill <numer>/pkill <nazwa>` – zakończ proces (na podstawie jego numeru lub nazwy, jeżeli nie chce się zakończyć możemy „zabić” dodając opcję `-9`)
- `vim/emacs/nano <plik>` – przykładowe edytory tekstu działające w terminalu
- `cat <plik>` – wypisuje zawartość pliku (bądź plików)
- `echo <coś>` – wypisuje coś na ekran
- `sudo/su` – uruchom jako inny użytkownik (bez podania dodatkowych opcji pozwala uzyskać uprawnienia superużytkownika root)
- `passwd` – zmień hasło
- `chown` – zmień właściciela pliku/katalogu
- `chmod` – zmień uprawnienia dostępu do pliku/katalogu
- `ssh <uzytkownik@zdalny.komputer>` – zdalna powłoka, pozwala zalogować się na innym komputerze
- `scp <plik_a> <plik_b>` – umożliwia kopiowanie plików między różnymi komputerami (plik na zdalnym komputerze podajemy w postaci `uzytkownik@zdalny.komputer:plik`, opcja `-r` pozwala skopiować katalog)
- `find <katalog>` – szuka plików w podanym katalogu (domyślnie wypisuje wszystkie, posiada liczne opcje filtrujące prezentowaną listę)
- `grep <coś> <plik>` – wypisuje linie zawierające coś nas interesującego w podanym pliku
- `clear` – wyczyść terminal
- `history` – historia poleceń (opcja `-c` czyści historię)

- **exit** – wyjście z powłoki
- **Ctrl+c** – przerwij bieżący proces (w razie niepowodzenia można też próbować bardziej radykalnego **Ctrl+**)
- **Ctrl+u** – skasuj właśnie wpisywaną w powłoce linię (przydatne gdy rozmyślimy się z wykonywania jakiegoś bardzo długiego polecenia)
- **Tab** – podpowiedz możliwe kontynuacje polecenia lub ścieżki

Struktura katalogów Linuxa

- **/** – katalog główny (czasem nazywany „root”, nie mylić z nazwą superużytkownika ani z jego katalogiem domowym)
- **/bin** – pliki wykonywalne
- **/boot** – pliki programu rozruchowego (w tym jądro Linuxa)
- **/dev** – urządzenia, Linux przedstawia je jako pliki, można z nich czytać i do nich zapisywać
- **/etc** – najważniejsze pliki konfiguracyjne
- **/home** – katalogi domowe użytkowników
- **/lib** – biblioteki, czasami występuje podział na 32- i 64-bitowe (np. dodatkowo jest katalog **/lib64**)
- **/mnt** – miejsce na dodatkowe punkty montowania dysków
- **/media** – punkty montowania nośników wymiennych (płyty, pendrive)
- **/opt** – miejsce na opcjonalne programy, tu trafiają niektóre aplikacje instalowane spoza oficjalnego repozytorium
- **/proc** – informacje o aktualnie działających procesach oraz systemie (w tym np. plik **/proc/cpuinfo** z informacjami o modelu procesora)
- **/root** – katalog domowy superużytkownika root
- **/run** – tymczasowe pliki obecnie działających procesów systemowych
- **/sbin** – pliki wykonywalne do administracji systemem (przeznaczone dla superużytkownika root)
- **/srv** – pliki udostępniane (serwowane) przez system (np. strony WWW)
- **/sys** – informacje o jądrze i urządzeniach, pozwala na zarządzanie pewnymi ich ustawieniami
- **/tmp** – pliki tymczasowe
- **/usr** – tzw. „drugorzędowa struktura plików”: niektóre foldery (np. **/bin** czy **/lib**) są tu powtórzone i zawierają pliki, które nie są kluczowe dla rozruchu systemu, a w niektórych nowych systemach wręcz to *te same pliki* – tzn. **/bin** jest dowiązaniem symbolicznym („skrótom”) do **/usr/bin**
- **/var** – pliki często ulegające zmianom (np. logi systemowe)

Uprawnienia dostępu do plików – jeżeli wywołamy polecenie **ls -l** by wypisać zawartość jakiegoś katalogu, łatwo zauważymy charakterystyczny ciąg liter i myślników, np.

```
-rw-r--r-- 1 pczachorowski pczachorowski 105239 Apr 26 12:01 notatki1.pdf
```

Kolejne pola (oddzielone spacjami) oznaczają: uprawnienia dostępu, liczbę dowiązań, właściciela pliku, grupę, rozmiar, datę ostatniej modyfikacji (w kilku polach) i – w ostatnim polu – nazwę pliku. Przede wszystkim interesuje nas tu pierwsze pole, a także trzecie i czwarte. W systemie Linux istnieją trzy kategorie uprawnień dostępu do pliku – odczyt, zapis i wykonywanie¹, oznaczane odpowiednio **r**, **w** i **x**. Odczyt i zapis raczej są dość oczywiste, wykonywanie dotyczy tego, czy plik można uruchomić (jak plik **.exe** lub skrypt). W przypadku katalogów odczyt pozwala na wypisanie zawartości, zapis – tworzenie i usuwanie plików, a wykonywanie

¹Istnieją też pojęcia tzw. **suid**, **guid** oraz **sticky bit**, które tu można spotkać oznaczone literami **s** i **t**, niemniej nie będziemy się nimi zajmować.

– czy do katalogu można wejść i uzyskać bardziej szczegółowe informacje o plikach. Uwaga – nawet bez uprawnienia do odczytu można otworzyć jakiś plik w tym katalogu jeżeli znamy jego dokładną nazwę a mamy prawa do odczytu dla pliku i wykonania dla katalogu.

Uprawnienia te można przydzielić osobno: właścicielowi pliku (który jest też wypisany w trzecim polu), grupie użytkowników (wypisana w polu czwartym, często będzie po prostu to jednoosobowa grupa zawierająca tylko właściciela pliku i o takiej samej nazwie, ale w ogólności daje to pewną swobodę w manipulowaniu uprawnieniami) oraz wszystkim pozostałym użytkownikom. To też jest wypisane przez polecenie `ls -l`, właśnie w tej kolejności: `rw`x właściciela, `rw`x grupy i `rw`x innych. Jeśli jakiegoś uprawnienia nie przyznano, w odpowiednim miejscu jest minus. Plik `notatki1.pdf` z przykładu może być czytany przez właściciela, grupę i kogokolwiek innego, modyfikowany (zapis) tylko przez właściciela, a uprawnień do wykonania nie ma nikt (dla pdfa są tu raczej niepotrzebne). Dodatkowe miejsce na samym początku (tutaj minus) oznacza typ: minus dla pliku, `d` dla katalogu (ang. *directory*) i `l` dla dowiązania (ang. *link*).

Dla kompletności wyjaśnijmy też pozostałe pola zwracane przez komendę `ls`. Drugie pole oznacza liczbę „twardych dowiązań” (ang. *hard links*) do danego pliku – te same dane fizycznie zapisane na dysku mogą być w systemie plików reprezentowane jako więcej niż jeden plik. Dla typowych plików zwykle będzie tu jedynka, dla katalogów – liczba podkatalogów plus dwa². Piąte pole zawiera rozmiar pliku w bajtach (możemy użyć `ls -lh` by dostać wygodniejsze jednostki).

Uprawnienia modyfikujemy poleceniem `chmod`: np. `chmod u+x,g-r,o+wx notatki1.pdf` doda uprawnienia wykonywania właścicielowi pliku (`u`), zabierze uprawnienia odczytu grupie (`g`) i doda uprawnienia zapisu oraz wykonywania pozostałym użytkownikom (`o`). Poleceniem `chown` z kolei możemy zmienić właściciela oraz grupę danego pliku: np. `chown root:www-data notatki1.pdf` zmieni właściciela na użytkownika (`root`), a grupę na (`www-data`). Teraz to ich będą dotyczyć odpowiednie uprawnienia. Polecenie `chown` wymaga podwyższonych uprawnień administracyjnych (o których dalej).

W domyślnych instalacjach systemu nierzadko zakłada się zaufanie między użytkownikami (bądź że z komputera i tak będzie korzystał tylko jeden użytkownik) i ich katalogi domowe mogą być przeglądane przez innych użytkowników niż właściciel. Można to łatwo ograniczyć, zabierając prawa czytania i wykonywania swojego katalogu (`chmod g-rwx,o-rwx ~`). Tylda oznacza katalog domowy użytkownika wywołującego polecenie. Po jego wykonaniu jedynie my (oraz użytkownik `root`, który może wszystko) będziemy mogli zaglądać do naszego katalogu domowego. Warto upewnić się, że nasz własny katalog na komputerach pracownianych jest tak właśnie zabezpieczony. Z drugiej strony niekoniecznie jest to zawsze pożądane – bo blokuje to dostęp również niektórym programom uruchomionym jako inny użytkownik³.

Superużytkownik – ta część notatek nie będzie zbyt przydatna dla Państwa na pracowni, gdyż nie nadano Wam podwyższonych uprawnień. Warto jednak się z nią zapoznać choćby po to, by zrozumieć, czemu niektóre działania mogą zakończyć się niepowodzeniem. Ponadto może to być cenna wiedza dla osób, które chciałyby zainstalować Linuxa na własnym sprzęcie.

W systemie Linux zwykły użytkownik jest objęty pewnymi ograniczeniami. Może jak najbardziej korzystać z większości programów i zarządzać swoimi własnymi plikami. Nie ma jednak uprawnień do ingerencji w ustawienia innych użytkowników i całego systemu, a także dostępu do odczytu niektórych danych. Nie ma np. prawa instalować, aktualizować i usuwać oprogramowania (chyba że poza menadżerem pakietów, na własny użytek – we własnym katalogu domowym),

²Raz od katalogu, w którym się znajduje, drugi od elementu `.`, kolejne od elementu `..` w każdym podkatalogu

³Np. serwer `www` – przez który moglibyśmy udostępniać zawartość katalogu `public.html` (widoczna jest pod adresem `studenci.fuw.edu.pl/~<nazwa_użytkownika>`).

zmieniać właściciela plików poleceniem `chown`, ani podglądać plików takich jak `/etc/shadow` (w którym zebrane są zaszyfrowane hasła wszystkich lokalnych użytkowników) i wiele logów systemowych. Do przeprowadzania tego typu operacji służy konto superużytkownika (`root`).

Do uzyskiwania supermocy `roota` służy polecenie `su` ⁻⁴. Po wywołaniu `su` zażądane od nas będzie hasło `roota`, zatem oczywiście musimy je znać.

W wielu systemach konto `root` nie ma ustawionego hasła, więc nie można się na nie zalogować w standardowy sposób (w tym poprzez `su`). Zamiast tego preferowane jest użycie polecenia `sudo` – np. `sudo su` by uzyskać to samo co poprzednio poleceniem `su` (również opcjonalny myślnik). Tutaj także żądane jest od nas hasło, ale – co kluczowe w tym całym dylemacie – jest to nasze własne hasło, a nie `roota`.

Obydwa podejścia mają swoje dobre i złe strony oraz zwolenników. Otóż np. korzystając z `sudo` uprawnień można udzielać różnym użytkownikom w różnym zakresie, nie tak zero-jedynkowo jak w przypadku `su` – gdy udostępniamy komuś hasło `roota` bądź nie. Jest to też wygodne gdy np. z jakiegoś powodu chcemy jednemu z użytkowników cofnąć uprawnienia administracyjne – wówczas nie musimy zmieniać hasła `roota` (i podawać go na nowo wszystkim innym adminom), wystarczy modyfikacja w pliku `/etc/sudoers` dotycząca tego jednego użytkownika.

Co więcej, świadomość, że w systemie jest konto o tak dużych uprawnieniach i standardowej nazwie może stanowić łakomy kąsek dla włamywaczy. Z drugiej strony, możliwość zdalnego logowania (przez `ssh`) jako `root` zwykle łatwo wyłączyć, zresztą od lat praktyką jest jej ograniczanie w już domyślnie instalowanych ustawieniach, więc użytkownik przeważnie nawet nie musi się tym sam zajmować. Możemy też (i powinniśmy) ustawić wystarczająco silne hasło `roota`, by włamanie w inny sposób (niż przez `ssh`) było bardzo trudne.

Warto mieć na uwadze, że jeśli korzystamy z `sudo` i mamy uprawnienia do administrowania systemem, a nasze hasło zostanie przez kogoś poznane, nie ma już żadnej dodatkowej warstwy ochronnej, jaką byłaby konieczność znajomości również hasła `roota`. Inna sprawa, że na prywatnym komputerze uzyskanie przez kogoś dostępu do naszych osobistych plików może być znacznie gorsze, niż zdobycie przez napastnika uprawnień admina.

Dlatego też pytanie czy `su` czy też `sudo` jest bardziej bezpieczne budzi od lat burzliwe dyskusje i jak widać tak naprawdę odpowiedź na nie jest dość sytuacyjna. W przypadku typowego osobistego komputera, z którego korzysta tylko jeden użytkownik i robi to świadomie (dba o zasady bezpieczeństwa, zachowuje ostrożność podczas ingerowania w ważne pliki systemowe itp.) nie jest to aż tak istotny wybór – po prostu warto wiedzieć o istnieniu obu tych mechanizmów i o tym, że różne systemy mogą domyślnie wybierać któryś z nich. Niezależnie od tego typu dywagacji należy przede wszystkim pamiętać, by używać konta `root` ostrożnie i odpowiedzialnie – czyli w szczególności uważnie czytać wywoływane polecenia, nie wykonywać nieznanych programów i ograniczyć operacje na koncie superużytkownika do niezbędnego minimum!

⁴Myślnik na końcu nie jest konieczny, ale bywa wygodny. Powoduje on, że zmienne środowiskowe ustawiane są w taki sposób, jakbyśmy faktycznie logowali się jako `root`. W praktyce m.in. sprawia to, że możemy łatwiej wywoływać niektóre przydatne `rootowi` programy (przez `program` zamiast `/sbin/program`). Przeniesieni też zostajemy do katalogu `/root`. W przypadku braku myślnika zachowywane są zmienne i katalog takie jakie były dotychczas (co też może być w pewnych sytuacjach korzystne).

Jak zacząć?

- Pracownia lub sala pracy własnej (1.23) – stacjonarnie.
- Zdalnie: <https://www.fuw.edu.pl/okwf-uslugi-stud.html>. Można zalogować się przez `ssh`⁵ na komputer `pracownia.okwf.fuw.edu.pl`. Jest on jednakże widoczny jedynie w wewnętrznej sieci wydziałowej, toteż najpierw prawdopodobnie trzeba zalogować się na (widoczny z zewnątrz) komputer „przesiadkowy” `tempac.fuw.edu.pl`.
- LiveUSB – popularne dystrybucje pozwalają wgrać obraz systemu na pendrive-a i uruchamiać system za jego pomocą, pozwala to na komfortowe testowanie bez większego ryzyka (chyba, że intencjonalnie zamontujemy partycje z dotychczasowym systemem). Szczegółowe instrukcje najlepiej czerpać ze strony konkretnej dystrybucji. Czasami w obiegu są także gotowe LiveDVD (dołączane np. do czasopism komputerowych).
- Instalacja na tym samym dysku co Windows – zmniejszyć partycję z systemem Windows i w wolnym miejscu utworzyć nową, na której zainstalujemy Linuxa (tworzeniem zajmie się już instalator Linuxa, ale zmniejszanie partycji z Windowsem najlepiej zrobić z poziomu Windowsa). Instalację przeprowadza się zwykle z poziomu LiveUSB/LiveDVD.
- Instalacja na dodatkowym dysku – bardzo bezpieczne i wygodne rozwiązanie. Nie trzeba wtedy w żaden sposób ingerować w istniejący system, przy starcie komputera będziemy musieli przytrzymać klawisz odpowiedzialny za pokazanie menu bootowania i wybrać Linuxa (chyba, że ktoś sobie jeszcze to uprości przez odpowiednie ustawienia w BIOS/UEFI).
- Windows Subsystem for Linux: <https://learn.microsoft.com/en-us/windows/wsl/install>. System Linux zostanie zainstalowany jako pewnego rodzaju maszyna wirtualna w istniejącym systemie Windows. Chyba najlepsza opcja dla osób, które nie chcą instalować całego nowego systemu, a jedynie mieć dostęp do programów używanych na pracowni. Instalacja jest wyjątkowo prosta i nieinwazyjna.

Pytania kontrolne

1. Jakiego polecenia powłoki użyć w systemie Linux do przeniesienia pliku `plik1.txt` w katalogu `katalog1` do katalogu `katalog2`?
2. W jakim katalogu w systemie Linux należy szukać pliku z hasłami użytkowników?
3. Co oznacza `drwxr-x---` wyświetlone przez polecenie `ls`?
4. Do czego służy komenda `ls /root`? Czemu gdy wywołamy ją jako zwykły użytkownik najprawdopodobniej uzyskamy informację o braku uprawnień? Co zrobić żeby polecenie to osiągnęło zamierzony skutek?

⁵W systemie Windows często proponuje się PuTTY lub inny podobny program, ale w nowych wersjach systemu można również po prostu doinstalować `ssh` (Ustawienia→System→Funkcje opcjonalne→Klient OpenSSH, a potem wywołujemy `ssh` w Wierszu poleceń)