

Technologie Informacyjne i Komunikacyjne R

Linux — ćwiczenia

Trzy zajęcia po 2 godziny, zadania 1–16

Na podstawie zestawu zadań: Katarzyna Grzelak (TIK-R, 2025)

Opracowanie: Michał Parniak (2026), mparniak@fuw.edu.pl

Zajęcia 1: pliki, katalogi, prawa dostępu, archiwa (zad. 1–3)

Pracujemy w katalogu `~/tiker`; wyniki zapisujemy w plikach, o które prosi treść zadania. Do opisu każdego polecenia można zajrzeć poleceniem `man polecenie` (wyjście klawiszem `q`) albo `polecenie --help`.

Przydatne polecenia

<code>pwd, ls -la, cd</code>	nawigacja	<code>chmod u+rwx, go-rwx</code>	prawa dostępu
<code>mkdir -p a/b</code>	tworzenie katalogów	<code>tar czvf a.tar.gz kat</code>	archiwum
<code>cp, cp -r, mv, rm</code>	kopiowanie, przenoszenie	<code>tar tzvf, tar xzvf</code>	lista / rozpakuj
<code>cat, less</code>	oglądanie plików	<code>zip -r, unzip</code>	archiwum zip

Na początek

- Zaloguj się, otwórz terminal. Sprawdź, w jakim katalogu jesteś (`pwd`), i obejrzyj jego zawartość, łącznie z plikami ukrytymi (`ls -la`).
- Znajdź skrót do katalogu roboczego `_work_`. Dokąd prowadzi? (`ls -l`)
- Przejdź do `/etc` i wróć do domowego trzema sposobami (`cd, cd ~, cd $HOME`). Wypróbuj klawisz `TAB` i strzałki.

Zad. 1 Kopiowanie z użyciem wzorców

W swoim katalogu domowym utwórz katalog `~/tiker/cwiczenia1` (jednym poleceniem, łącznie z katalogiem `tiker`). Przy pomocy **jednego** polecenia skopiuj do niego z katalogu `/proc` wszystkie pliki, które mają w nazwie ciąg liter `info`. Obejrzyj zawartość jednego z nich (np. `cpuinfo`): jaki procesor ma Twój komputer i ile ma rdzeni?

Wskazówka: `mkdir -p`, wzorec z gwiazdkami. Pliki w `/proc` są „wirtualne” — `ls -l` pokazuje rozmiar 0, choć mają treść.

Zad. 2 Kopiowanie katalogu i prawa dostępu

Z katalogu domowego użytkownika `kaste` (`~kaste/`) skopiuj do katalogu `~/tiker/cwiczenia1` katalog `sum` razem z całą zawartością. Zmień prawa dostępu tak, żeby cały katalog `sum` i wszystkie zawarte w nim pliki i podkatalogi mógł oglądać i zmieniać **tylko właściciel**. Sprawdź wynik poleceniem `ls -lR`.

Wskazówka: `cp -r, chmod -R`. Bez prawa `x` nie da się wejść do katalogu — co się stanie, gdy odbierzesz je sobie?

Zad. 3 Archiwa

Z katalogu `sum` (ze wszystkimi podkatalogami i plikami) utwórz:

- archiwum `sum.tar`,
- skompresowane archiwum `sum.tar.gz` (inna nazwa: `sum.tgz`),

c) archiwum `sum.zip`.

Porównaj rozmiary archiwów (`ls -lh`). Obejrzyj zawartość `sum.tar.gz` bez rozpakowywania. Następnie rozpakuj je w nowym katalogu `~/tikr/test` i sprawdź, czy zachowały się ustawione w zad. 2 prawa dostępu.

Wskazówka: `tar` z opcjami `c`, `z`, `t`, `x`, `v`, `f`; `zip -r`.

Zajęcia 2: konfiguracja, strumienie, potoki, procesy (zad. 4–9)

Pracujemy w katalogu `~/tikr`. Jeżeli zadania 1–3 z poprzednich zajęć nie są skończone, zacznij od nich — zad. 11 i 12 (zajęcia 3) korzystają z katalogu `sum` z zad. 2.

Przydatne polecenia

<code>du -s *</code> , <code>du -sh kat</code>	zajętość dysku	<code>ps -ef</code> , <code>kill -9 PID</code>	procesy
<code>sort -n</code> , <code>sort -h</code>	sortowanie	<code>find kat -name '*x*'</code>	szukanie po nazwie
<code>head -n 1</code> , <code>tail -n 1</code>	początek / koniec	<code>grep -rl tekst kat</code>	szukanie w treści
<code>cut -d: -f5</code>	wycinanie pól	<code>> >> 2> /dev/null</code>	przekierowania
<code>wc -l</code>	liczba linii	<code> </code>	potok

Zad. 4 Ile mam miejsca?

- Poleceniem `quota` sprawdź, ile miejsca masz do dyspozycji w przestrzeni dyskowej OKWF i ile już zajmujesz.
- Za pomocą poleceń `du`, `sort` i potoku `|` znajdź plik lub katalog, który zajmuje najwięcej miejsca w Twoim katalogu **domowym**, a potem to samo dla katalogu **roboczego**.

Wskazówka: `du -s *` podaje rozmiar każdego elementu katalogu; `sort -n` sortuje liczbowo. Pliki ukryte nie pasują do `*` — dodaj wzorzec `.[!..]*`.

Zad. 5 Plik `.bashrc` i aliasy

- Sprawdź, czy w katalogu domowym jest plik `.bashrc`. Jeśli jest, **najpierw zrób kopię**: `cp ~/.bashrc ~/.bashrc.kopia`. Dowolnym edytorem tekstu (`nano`, `code`, `gnome-text-editor`) dopisz na końcu `~/.bashrc` linię (jeśli pliku nie było — utwórz go):

```
| echo Witaj
```

Na końcu linii wciśnij ENTER, zapisz plik i otwórz nowy terminal. Co widzisz?

- Polecenie `alias` bez argumentów wypisuje zdefiniowane aliasy. Sprawdź, czy już jakieś istnieją.
- W terminalu zdefiniuj alias `alias dir='ls -la'` i sprawdź działanie `dir`. Czy działa w innym, nowo otwartym terminalu? Dlaczego?
- Zdefiniuj aliasy dla `rm`, `mv` i `cp` tak, aby przy ich wykonywaniu użytkownik był pytany o zgodę na usunięcie lub nadpisanie plików. Potrzebną opcję znajdź w `man rm` (i analogicznie dla `cp`, `mv`). Zdefiniuj też alias `gedit` dla edytora `gnome-text-editor`.
- Dopisz te aliasy do `~/.bashrc` (na końcu pliku ENTER) i usuń linię `echo Witaj`. Otwórz nowy terminal i sprawdź, czy aliasy są zdefiniowane.
- Plik `.bashrc` czyta każdy nowy terminal, ale przy logowaniu przez `ssh` `bash` czyta `~/.bash_profile`. Sprawdź (`grep bashrc ~/.bash_profile`), czy `.bash_profile` wczytuje `.bashrc`. Jeśli nie — dopisz do niego linię `source ~/.bashrc`.

Zad. 6 Największy element katalogu domowego

Zapisz do pliku `~/tikr/cwiczenia1/najwiekszy.txt` nazwę i rozmiar folderu (lub zwykłego pliku), który zajmuje najwięcej miejsca w Twoim katalogu domowym — jednym poleceniem, bez ręcznego przepisywania.

Wskazówka: `du`, `sort`, `tail` lub `head`, przekierowanie `>`. Opcja `-h` w `du` i `sort` daje rozmiary w kB/MB/GB.

Zad. 7 Procesy

- Polecenie `ps -ef` wypisuje wszystkie procesy działające na komputerze. Zapisz do pliku `procesy.txt` tylko **własne** procesy, korzystając wyłącznie z `ps -ef` i `grep`. Czy w wyniku są jakieś procesy, które nie są Twoje? Skąd się wzięły?
- Znajdź proces-rodzica procesu `ps` (kolumny PID i PPID). Co to za program? A kto jest jego rodzicem?
- Otwórz **drugi** terminal. Z pierwszego zamknij drugi, usuwając poleceniem `kill -9` proces powłoki `bash`, która w nim działa. (Numer procesu powłoki wypisuje `echo $$`.) Nie zabijaj procesu `gnome-terminal-server` — obsługuje on wszystkie okna terminala naraz.

Zad. 8 Pierwszy skrypt

Przy użyciu edytora tekstu utwórz w katalogu `~/tikr` skrypt `szukaj1` zawierający, linijka po linijce, polecenia wykonujące poniższe kroki. Wyjście diagnostyczne (`stderr`) wszystkich poleceń przekieruj do `/dev/null`.

- Utwórz katalog `~/tikr/cwiczenia2`.
- W katalogu `/etc` i wszystkich jego podkatalogach znajdź pliki, które mają w **nazwie** ciąg liter `root`, i zapisz ich listę do pliku `~/tikr/cwiczenia2/pliki1.txt`.
- W katalogu `/etc` i wszystkich jego podkatalogach znajdź pliki, które zawierają ciąg liter `root` w swojej **treści**, i zapisz ich listę (same nazwy plików) do `~/tikr/cwiczenia2/pliki2.txt`.
- Listę zawartości swojego katalogu domowego zapisz do `~/tikr/cwiczenia2/pliki3.txt`.
- Za pomocą `cat` połącz trzy powstałe pliki w jeden: `~/tikr/cwiczenia2/pliki.txt`.
- Bez użycia edytora dopisz na **początku** pliku `pliki.txt` liczbę jego linii.

Uruchom skrypt poleceniem `bash szukaj1` i sprawdź wyniki. Skrypt musi działać tak samo niezależnie od tego, w którym katalogu leży i z którego go uruchamiamy — sprawdź to, przenosząc go w inne miejsce.

Wskazówka: `find`, `grep -rl`, `wc -l`. Pamiętaj, że `cmd < plik > plik` kasuje plik — użyj pliku pomocniczego.

Zad. 9 Praca zdalna: ssh, wget i baza użytkowników

- Poleceniem `ssh` połącz się z komputerem `tempac.okwf.fuw.edu.pl`. Sprawdź, na jakim komputerze jesteś (`hostname`) i w jakim katalogu. Dalsze punkty wykonuj na `tempacu`.
- Klasyczna baza użytkowników ma format pliku `/etc/passwd`: jedna linia na osobę, pola oddzielone dwukropkami (`login`, `hasło`, `UID`, `GID`, *imię i nazwisko*, `katalog domowy`, `powłoka`). Obejrzyj `/etc/passwd` (`head /etc/passwd`). Konta studentów OKWF nie są tam zapisane — komputery pobierają je z serwera sieciowego, którego nie da się wylistować. Dlatego przygotowaliśmy **fikcyjną** bazę w tym samym formacie. Pobierz ją:

```
wget -O uzytkownicy.txt https://cloud.qodl.eu/s/gwFHig4crWRjEtW/download
```

Ile ma linii?

- c) Sprawdź, ilu użytkowników ma na imię (pierwsze lub drugie) tak jak Ty (imiona są zapisane bez polskich znaków). Jeśli nikogo nie ma, wybierz imię z bazy.
- d) Wypisz same imiona i nazwiska tych osób do pliku `nazwiska.txt`.
- e) * Ile osób ma **pierwsze** imię takie jak Ty?
- f) Jednym poleceniem zapisz do pliku `loginy.txt` posortowaną listę nazw użytkowników (loginów).

Na koniec wyloguj się poleceniem `logout`. Czy pliki utworzone na `tempacu` widać na komputerze w pracowni? Dlaczego?

Wskazówka: `cut -d: -f5, grep -w` (całe słowa — bez `-w` imię Jan znajdzie też Jankowskiego), `wc -l, sort`.

Zajęcia 3: skrypty (zad. 10–16)

Każdy skrypt zaczynamy linią `#!/bin/bash`, nadajemy mu prawo wykonywania (`chmod +x skrypt`) i uruchamiamy jako `./skrypt`. Skrypty powinny działać z **dowolnego katalogu** — sprawdź to. Gdy skrypt nie działa, uruchom go poleceniem `bash -x skrypt`, które wypisuje każde wykonywane polecenie. Zadania 15 i 16 są dodatkowe.

Przydatne konstrukcje

<code>\$1 \$2 \$3, \$#</code>	argumenty, ich liczba	<code>[-z "\$1"]</code> , <code>[-n "\$1"]</code>	pusty / niepusty napis
<code>\$(polecenie)</code>	wynik polecenia	<code>[-d kat]</code> , <code>[-e plik]</code>	katalog / plik istnieje
<code>read zmienna</code>	wczytanie z klawiatury	<code>[\$a -lt \$b]</code>	porównanie liczb
<code>\$(a * b)</code>	arytmetyka	<code>for i in \$(seq 1 \$n)</code>	pętla
<code>exit 1</code>	zakończenie z błędem	<code>case "\$x" in ... esac</code>	wybór

Zad. 10 Zmienne środowiska

Korzystając ze zmiennych środowiska napisz skrypt `info`, który po uruchomieniu wypisze:

```
Witaj KKK
Pracujesz na komputerze LLL
Używasz powłoki MMM
Aktualny katalog to NNN
```

gdzie KKK, LLL, MMM, NNN zostaną automatycznie zastąpione odpowiednio nazwą użytkownika, nazwą komputera, nazwą powłoki i nazwą bieżącego katalogu. Uruchom go w kilku katalogach. Dodaj czwartą linię z liczbą plików w bieżącym katalogu.

Wskazówka: `printenv` wypisuje zmienne środowiska; liczba plików: `$(ls | wc -l)`.

Zad. 11 Sklejanie wiersza

W plikach w katalogu `sum` (zad. 2) ukryte są zwrotki wiersza.

- a) Jednym, jak najkrótszym poleceniem połącz wszystkie zwrotki i zapisz je w pliku `wierszyk.txt`.
- b) * Kolejne zwrotki są w plikach nazwanych kolejnymi literami alfabetu (`a`, `b`, `c`, ...), rozrzuconych po różnych podkatalogach. Napisz skrypt, który skleja zwrotki we właściwej kolejności. Utwórz w nim katalog pomocniczy i za pomocą pętli `for` skopiuj do niego wszystkie pliki ze zwrotkami.

Wskazówka: najpierw obejrzyj strukturę katalogu: `ls -R sum` albo `find sum`.

Zad. 12 Wyszukiwanie z argumentami

- W plikach w katalogu **sum** (i podkatalogach) znajdź wszystkie wystąpienia ciągu liter **sum**, bez rozróżniania wielkich i małych liter. Wynik zapisz do pliku **sum.txt**.
- Napisz skrypt **szukaj2**, który znajduje wszystkie wystąpienia **dowolnego** ciągu liter w plikach w katalogu o **dowolnej** nazwie i zapisuje wynik do pliku o **dowolnej** nazwie. Wszystkie trzy nazwy są argumentami wywołania:

```
| ./szukaj2 sum ~/tikr/cwiczenia1/sum wynik.txt
```

Wyjście diagnostyczne przekieruj do **/dev/null**. Jeśli podano mniej niż trzy argumenty, skrypt wypisuje, jak go używać, i kończy działanie.

Zad. 13 Katalogi tylko wtedy, gdy ich nie ma

- Napisz skrypt, który tworzy katalog **zadanie**, w nim katalog **testy**, a następnie kopiuje do **zadanie/testy** plik **meminfo** z katalogu **/proc**. Przed tworzeniem każdego katalogu skrypt sprawdza, czy katalog istnieje, i tworzy go tylko wtedy, gdy nie istnieje. Uruchom skrypt dwa razy.
- Napisz skrypt, który za pomocą pętli **for** tworzy strukturę 5 zagnieżdżonych katalogów **test1/test2/test3/test4/test5**. Przed tworzeniem każdego katalogu sprawdza, czy on istnieje; jeśli tak — wypisuje stosowny komunikat.
- Zmień skrypt z punktu b tak, aby liczba katalogów była podawana jako argument wywołania.

Wskazówka: `[ -d kat ], [ ! -d kat ]`; `for i in $(seq 1 $n)`; ścieżkę można budować w zmiennej: `sciezka=$sciezka/test$i`.

Zad. 14 Kopiowanie plików o danym rozszerzeniu

Napisz skrypt **kopiuj**, wywoływany jako

```
| ./kopiuj katalog_docelowy katalog_zrodlowy rozszerzenie
```

który tworzy katalog o nazwie podanej jako pierwszy argument i kopiuje do niego z katalogu podanego jako drugi argument wszystkie pliki o rozszerzeniu podanym (bez kropki) jako trzeci argument, np. `./kopiuj kopia /etc conf`. Skrypt:

- sprawdza, czy podano pierwszy argument (test niepustego napisu); jeśli nie — wypisuje komunikat i kończy działanie;
- sprawdza, czy katalog docelowy istnieje; jeśli nie — tworzy go, jeśli tak — wypisuje komunikat, że podano nazwę istniejącego katalogu (pliki kopiuje wtedy do istniejącego katalogu).

Sprawdź działanie skryptu dla nazw ze spacją (`./kopiuj "moja kopia" /etc conf`).

Zadania dodatkowe

Zad. 15 Plik z aliasami

Napisz skrypt, który sprawdza, czy istnieje plik `~/aliasy.sh`.

- Jeśli nie istnieje: wypisuje o tym komunikat i tworzy go, wpisując do niego aliasy dla **rm**, **cp** i **mv** z zad. 5.
- Jeśli istnieje: wypisuje komunikat `Plik ~/aliasy.sh istnieje i zawartość pliku`.

Jak wczytać aliasy z tego pliku do bieżącego terminala?

Wskazówka: `here-document: cat << KONIEC > plik`.

Zad. 16 Kalkulator

Napisz skrypt `kalkulator` wykonujący proste działania na liczbach całkowitych. Skrypt prosi o podanie pierwszej liczby, drugiej liczby i znaku działania (+, -, * lub /), wykonuje działanie i wypisuje wynik. Dla nieznanego działania wypisuje komunikat o błędzie. Co się dzieje przy dzieleniu przez zero? Zabezpiecz skrypt przed tym przypadkiem.

Wskazówka: `read, $((a + b)), case`. Dlaczego znak * trzeba ująć w cudzysłów?