Spis treści

[**ls, typ pliku, cd, mkdir, mv, rm, touch, stat, less** 1](#_Toc157559969)

[**Linki, aliasy, help, man, whatis, info** 5](#_Toc157559970)

[**tar, gzip, bzip2, zip** 7](#_Toc157559971)

[**Git** 8](#_Toc157559972)

[**Wyrażenia arytmetyczne, cal, date, locate, updatedb** 14](#_Toc157559973)

[**Historia, script file** 15](#_Toc157559974)

[**WYRAŻENIA REGULARNE** 16](#_Toc157559975)

[**cut, paste, join, comm, diff, patch, tr** 17](#_Toc157559976)

[**Prawa dostępu** 19](#_Toc157559977)

[**bash** 21](#_Toc157559978)

[**test, pętle, here** 23](#_Toc157559979)

[**Parametry** 26](#_Toc157559980)

[**Markdown** 28](#_Toc157559981)

[**Katalogi** 29](#_Toc157559982)

[**Teoria git** 30](#_Toc157559983)

[**date** 31](#_Toc157559984)

[**Skróty klawiszowe** 32](#_Toc157559985)

# **ls, typ pliku, cd, mkdir, mv, rm, touch, stat, less**

#Ostatni znak promta #-superużytkownik (root), $zwykły

df – ilość wolnej przestrzeni na dyskac free - ilość wolnej pamięci

Koniec: exit, ctrl+D

katalog nadrzędny(macierzysty)

**ls – wylistowuje pliki i katalogi w bieżącym katalogu**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Parametr** | | | **Opis parametru** | **Typ elementu (pierwsza litera)** | **Opis elementu** |
| -a | | | Wyświetlenie wszystkich plików i katalogów wraz z plikami ukrytymi plus symbole . oraz .. | - | Zwykły plik |
| -A | | | Wyświetlenie wszystkich plików i katalogów wraz z plikami ukrytymi bez symboli . oraz .. | b | Specjalny plik blokowy |
| -B | | | Wyświetlenie wszystkich plików i katalogów bez ich kopii zapasowych (ze znakiem ~ na końcu nazwy) | c | Specjalny plik znakowy |
| -C | | | wylistowania wszystkich plików i folderów w aktualnym katalogu, wyświetlając je w kolumnach | d | Katalog |
| -I, --ignore=wzorzec | | | Wyświetlenie plików, które nie są zdefiniowany przez wzorzec | l | Dowiązanie symboliczne |
| wzorzec | | | Wyświetlenie elementów pasujących do wzorca | p | Nazwany potok |
| -R, --recursive | | | Rekurencyjne wyświetlenie zawartości katalogów i jego podkatalogów | s | gniazdo |
| -r, --reverse | Odwrócenie kolejności wyświetlania katalogów (alfabetycznie) | -h --human-readable | W połączeniu z opcją -l wyświetla wielkości plików i katalogów w formacie "zrozumiałym dla człowieka" czyli w kilobajtach/megabajtach/gigabajtach |
| -S, --sort=size | Posortowanie wyniku według wielkości plików | -i | pierwsze pole to numer inode | | |
| -t, --sort=time | Posortowanie wyniku według czasu ich modyfikacji | -F | Ta opcja doda znak wskaźnika na końcu każdej wylistowanej nazwy. Na przykład doda ukośnik (/), jeśli nazwa jest katalogiem. | | |
| -X, --sort=extension | Posortowanie wyniku według ich rozszerzeń | -d | Zwykle, jeśli określony jest katalog, ls wylistuje zawartość katalogu, a nie sam katalog. Użyj tej opcji razem z opcją -l, aby zobaczyć szczegóły katalogu, a nie jego zawartość | | |
| -u | | | Posortowanie wyniku według czasu ostatniego dostępu do pliku | | |

| **Typ elementu i prawa dostępu do niego** | **Liczba powiązań do tego elementu** | **Właściciel pliku** | **Grupa, która została przypisana do tego pliku** | **Rozmiar elementu** | **Data modyfikacji** | **Nazwa elementu** |
| --- | --- | --- | --- | --- | --- | --- |
| drwxrwxr-x | 3 | mateusz | mateusz | 4096 | wrz 26 16:12 | katalog |

ścieżka absolutna – od katalogu głównego

ścieżka względna – od katalogu bieżącego

cd Zmienia katalog roboczy na Twój katalog domowy.

cd - Zmienia katalog roboczy na poprzedni katalog roboczy i printuje ścieżkę absolutną

cd ~nazwa\_użytkownika Zmienia katalog roboczy na katalog domowy użytkownika o nazwie nazwa\_użytkownika.

|  |  |
| --- | --- |
| **mkdir –** tworzy katalog mkdir dir{1..100} mkdir dir{1..100} && touch dir{1..100}/{a..z}  mkdir {2007..2009}-{01..12} | |
| -m | Można nadać prawa dostępu do danego katalogu w momencie jego tworzenia. |
| -p | Utworzenie całej ścieżki katalogów |
| -v | Wyświetlenie informacji o tym, czy katalog został utworzony |

|  |  |
| --- | --- |
| **rmkdir – usuwa puste katalogi** | |
| --ignore-fail-on-non-empty | Użytkownik nie zostanie poinformowany o podjętej próbie usunięcia katalogu niepustego |
| -v, --verbose | Zostaniemy poinformowani o tym, że katalog został pomyślnie usunięty, lub że jest on nie pusty |
| -p, --parents | Usuwa katalog, wraz z jego pustymi podkatalogami |
| -R | usuwa katalog z dowolną zawartością |

| **Parametr cp** | **Opis parametru** |
| --- | --- |
| -a | Kopiuje pliki i katalogi wraz z wszystkimi ich atrybutami, w tym właścicielstwem i uprawnieniami. |
| -b [metoda], --backup [metoda] | Plik istniejący już w danej lokalizacji zostanie zapisany jako kopia zapasowa, a nie nadpisana przez kopiowany plik. |
| -d, --no-dereference | Kopiowanie dowiązań symbolicznych, na które te dowiązania wskazują. |
| -f, --force | Jeśli istnieje już plik o identycznej nazwie, to zostanie on nadpisany bez wyświetlenia jakiegokolwiek pytania. |
| -i, --interactive | Jeśli występuje element o identycznej nazwie, to system zapyta, czy chcemy go zastąpić plikiem, który kopiujemy. |
| -l, --link | Tworzenie dowiązania twardego |
| -p, --preserve | Zachowuje oryginalne prawa i wszelkie inne dane kopiowanego pliku. |
| -r | Kopiowanie rekurencyjne całych katalogów i ich zawartości |
| -R, --recursive | Działa podobnie jak -r, ale dodatkowo zachowuje wszelkie elementy niebędące plikami |
| -S przyr\_kopii, --suffix=przyr\_kopii | Wraz z parametrem -b, można zastosować własną nazwę suffiksów, a nie domyślnie znak tyldy |
| -u | tylko pliki, które nie istnieją lub są nowsze niż odpowiadające im pliki w katalogu docelowym. |
| -v, --verbose | Wyświetlenie informacji o każdym pliku, który został skopiowany |

**mv - służy zarówno do przenoszenia, jak i zmieniania nazwy plików**

mv NAZWA\_ELEMENTU1 NAZWA\_ELEMENTU2

mv NAZWA\_ELEMENTU1 NAZWA\_ELEMENTU2 NAZWA\_KATALOGU (katalog musi istnieć)  
parametry takie jak cp -u

-u, --update - Podczas przenoszenia plików z jednego katalogu do drugiego, przenosi tylko pliki, które nie istnieją lub są nowsze niż odpowiadające im pliki w katalogu docelowym. (parametry jak w cp)

| **rm – usuwa pliki i katalogi** (rm plik1 plik2 plik3) | |
| --- | --- |
| -r, -R, --recursive | Usuwanie rekurencyjne całych struktur na dysku |
| -f, --force | Usuwanie plików bez pytania o potwierdzenie i zgłaszania błędów |
| -i, --interactive | Wyświetlenie zapytania, czy należy usunąć dany element z dysku |
| -v, --verbose | Wyświetlenie informacji o pliku, który został przez nas usunięty |

|  |  |  |  |
| --- | --- | --- | --- |
| **cat – wyświetla zawartość pliku (odczytuje jeden lub więcej plików i kopiuje je na standardowe wyjście)** | | | |
| -A | Działa tak samo jak parametry -vET | -s | Usuwa zbędne puste wiersze, aby nigdy nie było dwóch pustych wierszy obok siebie |
| -b | Numeruje wszystkie niepuste wiersze podczas ich przekazywania do standardowego wyjścia | -t | Działa tak samo jak -vT |
| -E | Zaznacza koniec każdego wiersza za pomocą znaku dolara | -T | Wyświetla tabulatory jako ^I |
| -n | Numeruje wszystkie wiersze podczas ich przekazywania do standardowego wyjścia | -v | Wyświetla znaki Ctrl, stosując znaki z użyciem ^M oraz znaki z ustawionym tzw. ósmym bitem za pomocą M- |

|  |  |
| --- | --- |
| **touch** – nowy plik | |
| -a | Zmienia tylko czas ostatniego dostępu do pliku |
| -c | Jeśli plik nie istnieje, to nie zostanie on utworzony |
| -m | Zmienia czas ostatniej modyfikacji pliku na czas bieżący |
| -r plik\_wzorca | Pozwala na podanie pliku, którego czas ma zostać pobrany i ustawiony jako czas modyfikacji tego pliku, którego czas chcemy zmienić |
| -t czas | Pozwala na samodzielne ustalanie czasu modyfikacji z RR-MM-DD gg-mm |
| -b | Powoduje, że nazwy pliku nie są pokazywane |

Można też łączyć pliki. cat movie.mpeg.0\* > movie.mpeg  
Można też użyć cat jak nano. cat > plik.txt i wtedy wpisać tekst i kliknąć ctrl+d  
Można przekierować standardowe wejście cat < plik.txt

|  |  |
| --- | --- |
| **file – drukuje krótki opis zawartości pliku (określa typ pliku)** | |
| -i | Pokazuje otrzymany typ MIME danego pliku, czyli informację o tym, z jakim dokładnie typem elementu mamy do czynienia i jaki program może otworzyć ten element. |
| -z | Za jego pośrednictwem można zajrzeć do plików skompresowanych i przeanalizować pliki w nich zawarte |

**stat** - podaje informacje o pliku i systemie plików ( rozmiar pliku, uprawnienia dostępu, identyfikator użytkownika i identyfikator grupy, czas dostępu do pliku w czasie utworzenia). Może zaakceptować kilka nazw plików

|  |  |
| --- | --- |
| -f | informacje o systemie plików |
| -c lub –format | dostosowuje format wyjściowy |
| -print=(format) | możemy podać znaki końcowe do formatu |
| -t | pokazuje wszystkie sekwencje formatów |

**less** – przeglądanie plików tekstowych

less /etc/passwd (wyświetlenie dzięki less informacji

| **Skrót klawiaturowy** | **Opis działania** |
| --- | --- |
| <liczba\_wierszy>b | Przewinięcie do tyłu o liczba\_wierszy |
| <liczba\_wierszy>d | Przewinięcie do przodu o liczba\_wierszy |
| F | Przewinięcie do przodu. Po dotarciu do końca danych wejściowych to polecenie oczekuje na kolejne dane wejściowe, a następnie kontynuuje przewijanie. Wówczas należy użyć kombinacji CTRL+C w celu przerwania działania programu |
| h | Wyświetlenie podsumowania wszystkich dostępnych poleceń |
| <liczba\_wierszy>Enter | Przewinięcie do przodu o <liczba\_wierszy>. Wartością domyślną jest 1. |
| q | Wyjście z programu |
| <liczba\_wierszy>Spacja | Wyświetlenie kolejnych <liczba\_wierszy> tekstu. Naciśnięcie spacji spowoduje wyświetlenie kolejnego ekrany tekstu |
| /wyrażenie\_regularne | Przejście do pliku w poszukiwaniu wierszy zawierających dopasowane wyrażenie regularne |
| ! polecenie | Wykonanie polecenia powłoki wskazanej w zmiennej środowiskowej SHELL |

**type command** - wyświetla rodzaj polecenia

Aby określić dokładną lokalizację danego programu wykonywalnego, używane jest polecenie which (działa tylko dla programów wykonywalnych)

# **Linki, aliasy, help, man, whatis, info**

Tworzenie twardego linku: ln NAZWA\_ELEMENTU LINK

Tworzenie symbolicznego linku (miękki/symlink): ln -s NAZWA\_ELEMENTU LINK

**Aliasy**

alias name=’string’ (string to co name ma robić)

unalias – usuwa alias

Sam alias wypisze wszystkie aliasy

Aliasy znikają, gdy kończy się sesja powłoki.

**Help -** bash ma wbudowaną funkcję pomocy dostępną dla każdego z wbudowanych poleceń powłoki

help cd ls --help

**man** – służy do przeglądania podręcznika   
man 5 passwd wyświetli stronę podręcznika opisującą format pliku etc/passwd

Strony podręcznika: Tytuł (nazwa strony), Streszczenie składni polecenia, Opis celu polecenia, Wykaz i opis każdej z opcji polecenia

Układ podręcznika:

1. Polecenia użytkownika
2. Interfejsy programowania dla wywołań systemowych jądra
3. Interfejsy programowania do biblioteki C
4. Pliki specjalne, takie jak węzły urządzeń i sterowniki
5. Formaty plików
6. Gry i rozrywki, takie jak wygaszacze ekranu
7. Różne
8. Polecenia administracji systemem

| **Parametr** | **Opis parametru** |
| --- | --- |
| -a, --all | Wyświetla strony dla wszystkich sekcji podręcznika |
| -K słowo\_kluczowe | Wyszukanie podanego słowa\_kluczowego we wszystkich stronach podręcznika man |
| --apropos -k słowo\_kluczowe | Wyświetla nagłówki stron zawierających wskazane słowo\_kluczowe, co pozwala na przejrzenie listy poleceń (Pierwsze pole w każdym wierszu wyników to nazwa strony podręcznika, a drugie pole pokazuje sekcję) |
| --manpath=ścieżka -M ścieżka | Przeszukuje katalogi podane w ścieżce pod kątem stron podręcznika man. |
| -t, --troff | Formatuje stronę podręcznika do wydruku w drukarce PostScript |

**whatis** - wyświetla nazwę i jednolinijkowy opis strony podręcznika pasującej do określonego słowa kluczowego (whatis ls)

**info** - wyświetla dokładniejsze i bardziej aktualne informacje w przypadku niektórych poleceń niż polecenie man

|  |  |
| --- | --- |
| ? | Wyświetl pomoc dotyczącą poleceń |
| page up lub backspace | Wyświetl poprzednią stronę |
| page down lub spaceba | Wyświetl następną stronę |
| n | Następny - wyświetl następny węzeł |
| p | Poprzedni - wyświetl poprzedni węzeł |
| U | W górę - wyświetl węzeł nadrzędny aktualnie wyświetlanego węzła, zwykle menu |
| enter | Podążaj za hiperłączem w miejscu kursora |
| Q | Wyjdź |

|  |  |
| --- | --- |
| Klawisz | Opis klawisza |
| H lub ? | Wyświetlenie listy komend polecenia info |
| spacja | Przewijanie wyświetlanego tekstu |
| M | Po tym klawiszu następuje wyświetlenie menu lub lista dostępnych menu |
| Q lub ctrl + C | Zakończenie działania polecenia info |
| tab | Przejście do następnego odnośnika hipertekstowego |
| end | Przejście na koniec tego węzła |
| l | Opuszczenie okna pomocy |
| [ | Przejście do poprzedniego węzła dokumentu |
| home | Przejście na początek tego węzła |

# **tar, gzip, bzip2, zip**

**tar** jest klasycznym narzędziem do archiwizacji plików ("tape archive")  
.tar (zwykłe" archiwum) lub .tgz, (skompresowane gzipem)   
tar tryb[opcje] ścieżka...

|  |  |
| --- | --- |
| c | Utwórz archiwum z listy plików i/lub katalogów. |
| x | Rozpakuj archiwum. |
| r | Dołącz określone ścieżki na końcu archiwum. |
| t | Wylistuj zawartość archiwum. |
| f | określenie nazwy archiwum tar |
| v | (verbose) bardziej szczegółowe wylistowanie |

tar xf archive.tar ścieżka\_dostępu – wyodrębnia pojedynczy plik (może być kilka ścieżek dostępu) obsługuje je z opcją --wildcards.

gzip [plik] - kompresja bezstratna gunzip [plik] - dekompresja

|  |  |
| --- | --- |
| -c | Wypisuje wynik na standardowe wyjście, zachowując oryginalne pliki. |
| -d | Dekompresja. Powoduje, że gzip działa jak gunzip. |
| -f | Wymusza kompresję, nawet jeśli skompresowana wersja oryginalnego pliku już istnieje. |
| -h | Wyświetla informacje o użyciu programu. |
| -l | Wyświetla statystyki kompresji dla każdego skompresowanego pliku. |
| -r | Jeśli jeden lub więcej argumentów w linii poleceń to katalog, rekurencyjnie kompresuje pliki w nich zawarte. |
| -t | Testuje integralność skompresowanego pliku. |
| -v | Wyświetla szczegółowe komunikaty podczas kompresji. |
| -number | Ustala poziom kompresji. Liczba całkowita od 1 (najszybsza, najmniejsza kompresja) do 9 (najwolniejsza, największa kompresja). Wartości 1 i 9 można również wyrazić jako --fast i --best. Domyślna wartość to 6. |

bzip2 – to samo co gzip, **ale bez –r** Do bzip2 dołączony jest również program bzip2recover, który próbuje odzyskać uszkodzone pliki .bz2.

Zip prócz kompresowania służy jako narzędzie do tworzenia "pakietów" o formacie .zip

zip [options] zipfile files\_list unzip [options] zip\_file

|  |  |  |  |
| --- | --- | --- | --- |
| -d | Usuwa plik z utworzonego archiwum .zip | -t | Testuje, czy dany plik jest poprawnym archiwum .zip |
| -u | Aktualizuje plik w archiwum .zip. Opcja przydatna w przypadku aktualizacji zawartości pliku lub dodania nowego pliku do archiwum. | -l | Wyświetla pliki i katalogi z pliku .zip wraz z rozmiarem i datą utworzenia tych plików |
| -m | Usuwa oryginalne pliki po spakowaniu. Innymi słowy przenosi pliki do pliku .zip (podobnie działa narzędzie gzip lub bzip2) | -n | Zakazuje nadpisywanie plików podczas wyodrębniania plików .zip |
| -r | Spakowanie katalogu rekurencyjnie | -p | Wyświetla zawartość archiwum na standardowe wejście |
| -x | Wyklucza pliki podczas tworzenia pliku .zip | -c | Działa podobnie jak -p, ale wyświetla przed zawartością nazwę pliku |
| -v | Wyświetla tryb szczegółowy lub drukowanie informacji. | -sf | Pokazuje pliki oraz katalogi znajdujące się w pliku .zip |

# **Git**

* git config --global user.name "Mateusz Miotk"
* git config --global user.email [mateusz.miotk@ug.edu.pl](mailto:mateusz.miotk@ug.edu.pl)

Jeśli nie jest skonfigurowany, Git używa domyślnego edytora Vim. Jeśli chcesz użyć innego edytora tekstu, np. Emacs:

* **git config --global core.editor emacs**

Jeśli chcesz sprawdzić swoje ustawienia, możesz użyć polecenia **git config –list**

**git init** - Utworzy nowy podkatalog o nazwie .git, który zawiera wszystkie niezbędne pliki repozytorium — szkielet repozytorium Git

**git status** - określa, stan plików **git status –s** – uproszczony zapis git status

**git diff** – pokazuje, co zostało zmienione, ale jeszcze nie umieszczone w poczekalni

**git diff –cached** - wyświetla co zostało do tej pory przygotowane

.gitignore

* Puste linie lub linie zaczynające się od # są ignorowane.
* Można stosować standardowe wyrażenia regularne.
* Wzory można zakończyć ukośnikiem (/) aby określić katalog.
* Wzorzec można zanegować, rozpoczynając go od znaku wykrzyknika (!)

**git commit -v** – przypomnienie co zatwierdzamy

**git commit -a** - automatycznie umieszcza każdy plik, który jest już śledzony przed wykonaniem zatwierdzenia

**git commit –amend** - zatwierdzenie ponownie w przypadku błędu

**git rm -f** - Jeśli zmodyfikowaliśmy plik i dodaliśmy go już do indeksu, musimy wymusić usunięcie

**git rm --cached** usuwa z poczekalni a nie z katalogu

**git rm mv**

**git log** - wyświetla zmiany dokonane w tym repozytorium w odwrotnej kolejności chronologicznej

-2 - ogranicza wyjście tylko do dwóch ostatnich wpisów

--pretty=oneline/short/full/fuller - Ta opcja zmienia dane wyjściowe dziennika na formaty inne niż domyślne

--pretty=format

|  |  |
| --- | --- |
| **Opcja** | **Opis opcji** |
| %H | Kod zatwierdzenia |
| %h | Skrócony kod zatwierdzenia |
| %T | Kod drzewa |
| %t | Skrócony kod drzewa |
| %P | Kod korzenia (ojca) |
| %p | Skrócony kod korzenia (ojca) |
| %an | Imię i nazwisko autora |
| %ae | Adres email autora |
| %ad | Data autora |
| %ar | Względna data autora (w stosunku do bieżącego dnia) |
| %cn | Nazwa osoby zatwierdzającej |
| %ce | Adres email osoby zatwierdzającej |
| %cd | Data zatwierdzeania |
| %cr | Względna data zatwierdzenia |
| %s | Tytuł zatwierdzenia |

|  |  |
| --- | --- |
| **Opcja** | **Opis opcji** |
| -p | Pokaż łatkę wprowadzoną przy każdym zatwierdzeniu. |
| --stat | Pokaż statystyki dla każdego zmodyfikowanego pliku przy zatwierdzeniu |
| --shortstat | Pokaż wyłącznie zmienione/wstawione/usunięte linię z opcji --stat |
| --name-only | Pokaż listę plików zmodyfikowanych po informacji o zatwierdzeniu. |
| --name-status | Pokaż również listę plików, których dotyczą dodane/zmodyfikowane/usunięte linie. |
| --abbrev-commit | Pokaż pierwsze znaki kodu zatwierdzenia |
| --relative-date | Wyświetlaj datę w formacie względnym (na przykład „2 tygodnie temu”) zamiast używać pełnej daty. |
| --graph | Wyświetl wykres ASCII gałęzi i historii repozytorium |
| --pretty | Pokaż zatwierdzenia w innym formacie. |
| --since i –until  git log --since=2.weeks | od kiedy do kiedy |
| --grep | pozwala na wyszukiwanie słów kluczowych w wiadomościach o zatwierdzeniu |
| --author | umożliwia filtrowanie według określonego autora |
| --all-match | Jeśli chcemy określić zarówno autora, jak i opcje grep, musimy dodać |
| -S | pobiera ciąg i pokazuje tylko zatwierdzenia, które wprowadziły zmianę w kodzie, który dodał lub usunął ten ciąg |

git restore --staged <plik> - aby wycofać zmiany z poczekalni

git restore <plik> - wycofuje zmiany

Zdalne repozytoria to wersje projektu, które są hostowane w Internecie lub sieci.

**git remote** - zawiera listę skróconych nazw każdego określonego zdalnego repozytorium

**git remote -v** - pokazuje adresy URL

**git remote add [shortname] [url]** - dodaje nowe zdalne repozytorium

**git fetch** - pobiera wszystkie informacje z gita, których nie mamy u siebie

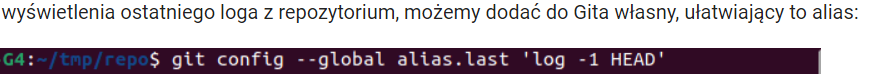
**git remote show [remote-name]** - Jeśli chcemy zobaczyć więcej informacji o konkretnym zdalnym repozytorium

**git remote rename git remote rm**

**git tag** - wyświetlanie wszystkich dostępnych tagów **git tag -l ‘v1..8.1’** – szukanie tagu o konkretnym wzorcu

**git show** - Dane etykiety wraz z tagowaną rewizją

**git push origin [nazwa-etykiety]** **git push origin master –tags**



**Gałąź** w Gicie jest po prostu lekkim, przesuwalnym **wskaźnikiem** na któryś z owych zestawów zmian

**git branch [nazwa]** + **git checkout [nazwa]** = **git checkout -b** – tworzy nową gałąź i jednocześnie przenosi się na nią

**git branch -d** – usuwa gałęź

**git mergetool** - uruchamia narzędzie wizualnego scalania i przeprowadza przez konflikty

**przekierowania, sort, uniq, wc, grep, head, tail, tee, echo, sleep, find**

„>” nazwa pliku - aby przekierować standardowe wyjście do innego pliku zamiast na ekran

plik docelowy zawsze jest zapisywany na początku

>> dopisuje do pliku zamiast go nadpisywać

Deskryptor pliku 2 – standardowy błąd Deskryptor pliku 1 – standardowe wyjście

Dopisujemy 2 przed > żeby przekierować też błąd Przechwytywanie całego wyjścia: 1) 2>&1 na końcu linii 2) &>

Aby tłumić komunikaty o błędach z polecenia, robimy to tak: ls -l /bin/usr 2> /dev/null

Za pomocą operatora potoku | standardowe wyjście jednego polecenia można przekierować do standardowego wejścia innego. polecenie1 | polecenie2

**sort**

|  |  |
| --- | --- |
| -o filename --output=filename | zapisujemy wynik polecenia do pliku o nazwie filename (tak samo jak przekierowanie) |
| -r --reverse | służy do odwracania porządku w sortowaniu |
| -n --numeric-sort | sort sortuje dane numerycznie, a nie leksykograficznie |
| -k <liczba>n | Sortuje dane według kolumny podane jako argument liczba |
| -c | sprawdza czy dany plik jest poprawnie posortowany |
| -u | Sortuje dane jednocześnie usuwając duplikaty |
| -M | Sortuje dane według miesiąca ze względu na używany język regionalny |
| -f --ignore-case | gnoruje podczas sortowania duże oraz małe litery traktując je tak samo |
| -b --ignore-leading-blanks | sort ignoruje początkowe spacje w wierszach i oblicza sortowanie na podstawie pierwszego niebiałego znaku w wierszu |
| -m | sortowania poprzez iloczyn kartezjański plików |
| -t char --field-separator=char | Parametr ten definiuje separator, jaki ma być użyty |

**uniq** przyjmuje posortowaną listę danych z wejścia standardowego lub z pojedynczego argumentu w postaci nazwy i domyślnie usuwa wszelkie duplikaty z listy

|  |  |
| --- | --- |
| -c --count | Wyświetli listę zduplikowanych linii wraz z liczbą powtórzeń |
| -d | wyświetla duplikaty |
| -D | wyświetla duplikaty |
| -u --unique | Wyświetla wyłącznie unikalne linię |
| -f n --skip-fields=n | ignorowane jest n pól w każdym wierszu |
| -s n | Działa podobnie jak -f n ale, ignoruje n znaków zamiast pól |
| -w n | możemy podać ograniczenie porównania do zadanej liczby znaków |
| -i --ignore-case | ignorowane są wielkie i małe litery |
| -z | kończy linię wartością NULL |

**wc** (word count) - wyświetlania liczby linii, słów i bajtów zawartych w plikach

|  |  |
| --- | --- |
| -l | liczba linii |
| -w | liczba słów |
| -c | liczba bajtów |
| -m | liczba znaków |
| -L | wartość najdłuższej linii pod względem znaków |

**grep** - to program służący do wyszukiwania wzorców tekstowych w plikach (grep wzorzec plik)

grep[ opcje ] wzorzec [lista-plikow]

|  |  |
| --- | --- |
| -c | Wyświetla jedynie ilość wierszy zawierających dopasowanie w każdym pliku |
| -C n | Wyświetla n wierszy kontekstu wokół każdego dopasowanego wiersza. |
| -f plik | Odczytuje plik zawierający po jednym wzorcu w każdym wierszu, a następnie w danych wejściowych wyszukuje wiersze zawierające dopasowania do wskazanych wzorców |
| -h | Powoduje, że w przypadku przeszukiwania wielu plików na początku żadnego wiersza nie będzie wyświetlona nazwa pliku. |
| -i | Powoduje, że małe litery we wzorcu będą dopasowywały wielkie litery w pliku oraz na odwrót |
| -l | Wyświetla jedynie nazwę katalogu pliku zawierającego jedno lub więcej dopasowań. Nazwa pliku będzie wyświetlona tylko jednokrotnie, nawet jeśli plik zawiera więcej niż jedno dopasowanie |
| -m n | Zatrzymuje odczyt każdego pliku lub standardowego wejścia po wyświetleniu n wierszy zawierających dopasowanie. |
| -n | Na początku każdego wiersza zostanie umieszczony numer danego wiersza w pliku. Sam plik nie musi zawierać numerów wierszy. |
| --quiet lub --silent -q | Żadne dane nie będą przekazywane do standardowego wyjścia, nastąpi jedynie ustawienie kodu wyjścia |
| -R | Opcja powoduje rekurencyjną analizę katalogów wymienionych na liście plików do przetworzenia. |
| -s | Ta opcja powoduje, że jeżeli plik wymieniony na liście-plików nie istnieje lub nie można go odczytać, to nie będą wyświetlane komunikaty błędów. |
| -v | wydrukuje tylko te linie, które nie pasują do wzorca |
| -w | Ta opcja powoduje, że wzorzec musi dopasować całe słowo. Omawiana opcja okazuje się użyteczna w przypadku szukania określonego słowa, które może występować również jako podciąg tekstowy w innym słowie pliku. |
| -x | Opcja powoduje, że wzorzec dopasowuje jedynie całe wiersze. |
| -o | Wyświetla tylko pasujące (niepuste) części pasującego wiersza, każda w osobnym wierszu. |

|  |  |  |  |
| --- | --- | --- | --- |
| opcja | grep | egrep lub grep -E | fgrep lub grep -f |
| wzorzec | zwykłe wyrażenie regularne | rozszerzone wyrażenie regularne | stały ciąg znaków |

find - przeszukuje dany katalog (i jego podkatalogi) w poszukiwaniu plików na podstawie różnych atrybutów

|  |  |
| --- | --- |
| -name | szuka po nazwie |
| -depth | powoduje przetwarzanie zawartości katalogu przed sprawdzeniem samego katalogu |
| -maxdepth głębokość | określa maksymalne zagnieżdżenie |
| -mindepth głębokość | określa minimalne zagnieżdżenie |
| -amin n | Pozwala wyszukać pliki, do których dostęp nastąpił n minut temu |
| -atime n | Dostęp do wyszukiwanego elementu nastąpił n \* 24 godziny temu |
| -cmin n | Wyszukuje pliki, których status został zmieniony n minut temu |
| -ctime n | Wyszukuje pliki, których status został zmieniony n \* 24 godziny temu |
| -cnewer file | dopasuj pliki lub katalogi, których zawartość lub atrybuty zostały ostatnio zmodyfikowane później niż w file |
| -empty | wyszukuje pliki puste |
| -group nazwa | Wyszukuje pliki należące do danej grupy |
| -iname wzorzec | Wyszukuje pliki, których nazwy pasują do wzorca, jednak nie uwzględnia różnicy liter w nazwach elementów |
| -links n | Wyszukuje pliki mające n dowiązań |
| -mmin n | Wyszukuje pliki modyfikowane n minut temu |
| -mtime n | Wyszukuje pliki modyfikowane n \* 24 godziny temu |
| -nouser | Wyszukuje pliki, do których nie pasuje żaden użytkownik |
| -nogroup | Wyszukuje pliki, do których nie została przypisana grupa |
| -perm prawa | Wyszukuje pliki lub katalogi, dla których określono odpowiednie prawa dostępu |
| -perm -prawa | Wyszukuje pliki lub katalogi, dla których wszystkie bity praw zostały ustawione |
| -perm +prawa | Wyszukuje pliki, w których jakikolwiek z bitów praw jest ustawiony |
| -size n[ckMG] | Wyszukuje pliki o określonym rozmiarze. Jeśli wielkość podawana przez n jest wyrażona w 512-bajtowych blokach. + oznacza więcej niż - mniej niż. Ponadto c oznacza pomiar plików w znakach, k w kilobajtach, M w megabajtach, a G w gigabajtach |
| -type - typ\_pliku | Wyszukuje elementy należące do określonego typu |
| -user nazwa | wyszukuje po właścicielu |
| -and lub -a lub nic | Dopasuj, jeśli testy po obu stronach operatora są prawdziwe |
| -not |  |
| -delete | Usuwa znalezione dopasowania |
| -ls | Wykonuje odpowiednik ls -dils na pasującym pliku. Dane wyjściowe są wysyłane na standardowe wyjście |
| -print | Wypisuje pełną ścieżkę do pasującego pliku na standardowe wyjście. Jest to akcja domyślna, jeśli nie określono żadnej innej akcji. |
| -quit | Zakończ, jeżeli zostało dokonane dopasowanie |
| -ok | działa podobnie do opcji -exec, ale przed wykonaniem komendy pyta o potwierdzenie |

find /ścieżka/do/katalogu -exec command {} \;

/ścieżka/do/katalogu: Określa katalog, w którym ma być wykonane przeszukiwanie.

-exec command {} \;: Wskazuje, że komenda command ma być wykonana dla każdego znalezionego pliku lub katalogu.

{} to specjalne miejsce, gdzie zostaną podstawione nazwy znalezionych plików. \; oznacza koniec polecenia -exec.

find /ścieżka/do/katalogu -type f | xargs ls xargs przekazuje listę do polecenia ls

Polecenie find udostępnia akcję -print0, która generuje dane wyjściowe oddzielone wartościami null, a polecenie xargs ma opcję --null (lub -0), która akceptuje dane wejściowe oddzielone wartościami null

xargs –show-limits – sprawdza maksymalny rozmiar wiersza poleceń

typy plików

|  |  |
| --- | --- |
| b | plik blokowy |
| c | plik specjalny |
| d | katalog |
| p | nazwany potok |
| f | zwykły plik |
| l | dowiązanie symboliczne |
| s | gniazdo |

**head** – drukuje pierwsze 10 linii **tail** – drukuje ostatnie 10 linii

|  |  |  |  |
| --- | --- | --- | --- |
| -n | x linii | | te same parametry co head + -f |
| -c | x znaków | -f | kontynuuje monitorowanie pliku, a gdy dodane są nowe linie, natychmiast pojawiają się na ekranie (ctrl+c przerywa) |
| -q | W przypadku wielu plików wyświetli najpierw pierwszych 10 linii z pierwszego pliku, potem z drugiego i tak dalej. |

**tee** - czyta standardowe dane wejściowe i kopiuje je do standardowego wyjścia i do jednego lub kilku plików (-a nie nadpisuje)

**echo** - wyświetla jej tekstowe argumenty na standardowym wyjściu

\ blokuje działanie specjalnych znaków

|  |  |
| --- | --- |
| **Sekwencja ucieczki** | **Znaczenie** |
| \a | Dzwonek (alarm powodujący pikanie komputera) |
| \b | Backspace |
| \n | Nowa linia; w systemach Unix-like powoduje to przesunięcie linii |
| \r | Powrót karetki |
| \t | Tabulator |

echo -e – pozwala na interpretację sekwencji ucieczki. Można je również umieścić wewnątrz $' '

**sleep** - czeka określoną liczbę sekund, a następnie kończy działanie

rozszerzeniem ścieżki dostępu - mechanizm, dzięki któremu działają symbole wieloznaczne

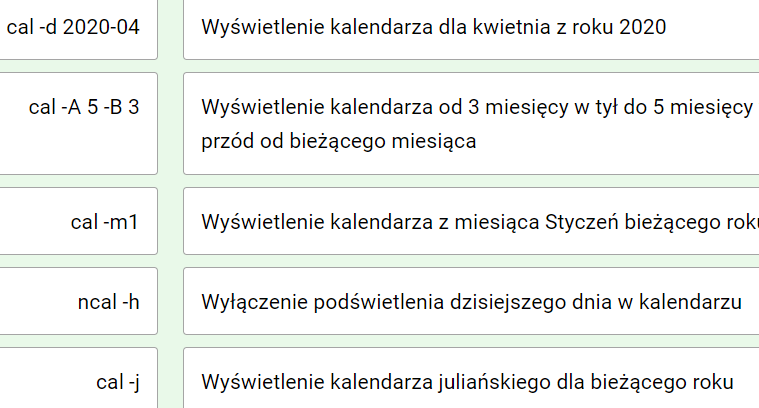
# **Wyrażenia arytmetyczne, cal, date, locate, updatedb**

$((wyrażenie)) Rozszerzenie arytmetyczne obsługuje tylko liczby całkowite

| **Notacja** | **Opis** |
| --- | --- |
| number | Domyślna notacja; system dziesiętny |
| 0number | System oktalny (ósemkowy) |
| 0xnumber | System szesnastkowy |
| base#number | System wskazany w base |

**cal** – kalendarz

| **Parametr** | **Opis parametru** |
| --- | --- |
| -j | Wyświetlona zostanie liczba dni od dnia pierwszego stycznia bieżącego roku |
| -y | Wyświetlenie całego kalendarza na bieżący rok |
| <rok> | Wyświetlenie dowolnych lat w formie kalendarza |
| <miesiac> <rok> | Wyświetlenie kalendarza dla danego miesiąca danego roku |
| -3 | Wyświetlenie kalendarza poprzedniego, bieżącego i następnego miesiąca |
| -A <liczba> | Wyświetlenie kalendarza bieżącego miesiąca i <liczba> następnych miesięcy |
| -B <liczba> | Wyświetlenie kalendarza bieżącego miesiąca i <liczba> poprzednich miesięcy |



ncal – lepszy kalendarz

%\_ date dopełni pole spacjami

%- date nie dopełni pola żadnymi znakami (wyrówna je do lewej strony)

locate – Program, który wykonuje szybkie wyszukiwanie nazw ścieżek w bazie danych (nowsze to slocate, mlocate)

updatedb – aktualizuje bazę danych (jako super użytkownik root)

# **Historia, script file**

history | less wyświetlić zawartość listy historii poleceń (kiedyś 500 teraz 1000)

Pierwsza liczba - numer linii polecenia

!n – przenosi do n linii w historii

Aby rozpocząć inkrementalne wyszukiwanie, naciśnijmy ctrl-R, a następnie wprowadźmy szukany tekst. Gdy go znajdziemy, możemy albo nacisnąć enter, aby wykonać polecenie, albo naciskając ctrl-J, skopiować linię z listy historii do bieżącej linii poleceń. Aby znaleźć następne wystąpienie tekstu (przesuwając się "w górę" listy historii), naciśnijmy ponownie ctrl-R. Aby zakończyć wyszukiwanie, naciśnijmy ctrl-G lub ctrl-C

skróty używane do manipulowania listą historii

|  |  |
| --- | --- |
| Kombinacja klawiszy | Akcja |
| ctrl-P | Przejdź do poprzedniego wpisu w historii. To samo co strzałka w górę. |
| ctrl-N | Przejdź do następnego wpisu w historii. To samo co strzałka w dół. |
| alt-< | Przejdź na początek (górę) listy historii. |
| alt-> | Przejdź na koniec (dół) listy historii, czyli bieżącą linię poleceń. |
| ctrl-R | Odwrotne inkrementalne wyszukiwanie. Szuka inkrementalnie od bieżącej linii poleceń w górę listy historii. |
| alt-P | Odwrotne wyszukiwanie, nieinkrementalne. Po naciśnięciu tego klawisza wprowadź ciąg wyszukiwania i naciśnij enter przed przeszukaniem. |
| alt-N | Wyszukiwanie do przodu, nieinkrementalne. |
| ctrl-O | Wykonaj bieżący element na liście historii i przejdź do następnego. Jest to przydatne, jeśli próbujesz ponownie wykonać sekwencję poleceń na liście historii. |

|  |  |
| --- | --- |
| Ciąg | Akcja |
| !! | Powtórz ostatnie polecenie. Prawdopodobnie łatwiej jest nacisnąć strzałkę w górę i enter. |
| !numer | Powtórz element listy historii o numerze. |
| !ciąg | Powtórz ostatni element listy historii zaczynający się od ciągu. |
| !?ciąg | Powtórz ostatni element listy historii zawierający ciąg. |

script file - rejestrowania całej sesji powłoki i przechowywania jej w pliku. Jeśli nie zostanie określony żaden plik, używany jest plik typescript

# **WYRAŻENIA REGULARNE**

? - zastępuje jeden dowolny znak

\*- zastępuje dowolną ilością znaków

[ ] - dopasowuje konkretne znaki np. f[ab] => fa i fb

[6-8] - myślnik oznacza przedział

! lub ^ zaraz po [ - negacja

REGEX

^ $ . [ ] { } - ? \* + ( ) | \ - metaznaki, trzeba cytować np. \.

separator np. /cos/

. - zastępuje dowolny znak na danej pozycji

|  |  |  |  |
| --- | --- | --- | --- |
| \* | 0 lub więcej | [\b] | backspace |
| .\* | 0 lub więcej dowolnych znaków | \f | nastepna strona |
| ^ | na początku wiersza | \n | znak nowej linii |
| $ | na końcu wiersza | \r | znak karetki |
| + | 1 lub więcej | \t | tabulator |
| ? | 0 lub 1 | \v | tabulator pionowy |
| | | lub | \b | granica słowa |
| {n} | znak występuje n razy | \B | przeciwieństwo granicy słowa |
| {n, m} | od n do m razy |  |  |
| {n,} | n lub więcej razy |  |  |
| {,m} | max m razy |  |  |

metaznaki szukają najdłuższego dopasowania, leniwe metaznaki jak najmniejszego

\*? +? {n,}? - metaznaki po dodaniu ? są leniwe

^[1-9][0-9]\*$ - liczba naturalna bez zera

^[0-9]\*$ - liczba naturalna

[aeiouy] – samogłoski

[ ! $# -eq 2 ]

[[ "$napis" =~ ^-?([0-9]+(\.[0-9]+)?|\.[0-9]+)$ ]] czy liczba

# **cut, paste, join, comm, diff, patch, tr**

**cut [option] [file] paste [option] [files]**

|  |  |
| --- | --- |
| -b | Wypisuje określoną wcześniej liczbę bajtów. |
| -c | Wyświetla określoną wcześniej liczbę znaków. |
| -f | Pozwala wyodrębniać dane bazując na kolumnach pliku |
| --complement | Wyświetla dopełnienie wybranego w poleceniu cut zbioru |
| -d --output-delimeter=STR | -d określa separator wejścia  --output-delimiter= ustawia separator wyjścia |

|  |  |
| --- | --- |
| -d | Określa separator, jeśli jako separator określono więcej niż jeden znak, to polecenia używa go w sposób cykliczny dla każdej linii pliku. |
| -s | Łączy pliki sekwencyjnie. Czyli czyta on wszystkie wiersze z jednego pliku i łączy je w jeden wiersz, z którego każdy oddzielony jest znakiem tabulatora |

**join [option] file1 file 2**

|  |  |
| --- | --- |
| -a [1|2] | Wypisuje również nie pasujące linie z pliku [1|2] |
| -v [1|2] | Drukuje wyłącznie nie pasujące linię z pliku podanego podobnie |
| -[1|2] n | Wskazuje inne niż pierwsze wspólne pole łączonych plików. Należy jednak po podaniu parametru -1 (co odpowiada pierwszemu plikowi) wskazać wspólną kolumnę. |
| -i | Ignoruje wielkość liter. |
| --nocheck-order | Domyślnie polecenie join sprawdza czy podane dane są posortowane (pod względem klucza). Aby tego nie robić należy użyć właśnie tego parametru. |
| -t | Określamy separator, który używamy do rozdzielenia pól. |

**comm [option] file1 file -** generuje trzy kolumny

|  |  |  |
| --- | --- | --- |
| Unikalne dla pierwszego pliku | Unikalne dla drugiego pliku | Wspólne dla obu |

|  |  |
| --- | --- |
| -[1|2|3] | nie wypisuje danej kolumny |
| -nocheck-order | Nie sprawdza czy pliki wejściowe są posortowane. |
| --output-delimiter=STR | Zmienianie separatora |

**diff [options] file1 file2**

| **Zmiana** | **Opis zmiany** |
| --- | --- |
| r1ar2 | Dołącz linie na pozycji r2 w drugim pliku do pozycji r1 w pierwszym pliku |
| r1cr2 | Zmień linie na pozycji r1 w pierwszym pliku na pozycje r2 w drugim pliku |
| r1dr2 | Usuń linie z pierwszego pliku na pozycji r1, który pojawiłby się na pozycji r2 w drugim pliku |

* c

| **Znak** | **Znaczenie** |
| --- | --- |
| pusto | Nie ma różnicy między plikami |
| - | Linia do usunięcia w pierwszym pliku |
| + | Linia do dodania w drugim pliku |
| ! | Zmieniona linia obydwu plikach |

* u -- Wyświetla różnice w trybie ujednoliconym.
* i – ignoruje wielkość liter

***patch*** służy do wprowadzania zmian w plikach tekstowych. Przyjmuje dane wyjściowe z polecenia diff i jest zwykle używany do konwertowania plików starszych wersji do nowszych wersji.

**diff -u plik1 plik2 > d.patch**

**patch < d.patch**

|  |  |
| --- | --- |
| -b | Dokonuje zachowania kopii zapasowej przed dokonaniem zmian dodając rozszerzenie .orig |
| -V numbered | Pozwala na dokonanie rejestracji wersji kopii zapasowej. |
| --dry-run | Służy do walidacji pliku otrzymanego z polecenia diff |
| -R | Wprowadza poprawki w odwrotnej kolejności. |

**tr [option] SET1 [SET2]**

|  |  |
| --- | --- |
| -c | Dopełnia do wskazanego zbioru w poleceniu |
| -d | Usuwa znaki |
| -s | Zastępuje powtarzające się znaki jednym znakiem. |

Klasy znaków przy poleceniu tr

|  |  |
| --- | --- |
| **Klasa znaków** | **Opis** |
| [:alnum:] | litery i cyfry - [A-Za-z0-9]. |
| [:word:] | [:alnum:] ze znakiem podkreślenia \_. |
| [:alpha:] | litery – [a-zA-Z] |
| [:cntrl:] | Kody kontrolne ASCII. Obejmuje znaki ASCII od 0 do 31 i 127. |
| [:blank:] | Zawiera spacje i znaki tabulacji – [\s\t] |
| [:digit:] | cyfry - [0-9]. |
| [:graph:] | Widoczne znaki. W ASCII zawiera znaki 33 do 126. |
| [:lower:] | małe litery - [a-z]. |
| [:punct:] | Znaki interpunkcyjne - [-!"#$%&'()\*+,./:;<=>?@[\\\]\_`{|}~] |
| [:print:] | Znaki pisane. To samo co [:graph:] plus znak spacji. |
| [:space:] | Białe znaki, w tym spacja, tabulator, powrót karetki, znak nowej linii, tabulator pionowy i wysuw strony  [\t\r\n\v\f] |
| [:upper:] | wielkie litery - [A-Z]. |
| [:xdigit:] | System szesnastkowy - [0-9A-Fa-f] |

$1 – informacje o tożsamości  
uid – identyfikator użytkownika (numer nadany przy tworzeniu konta), który jest mapowany na nazwę użytkownika dla ludzi

gid (identyfikator grupy) – przypisany użytkownikowi

Konta użytkowników są definiowane w pliku /etc/passwd, a grupy są definiowane w pliku /etc/group. etc/shadow zawiera informacje o haśle użytkownika.

Dla każdego konta użytkownika plik /etc/passwd definiuje nazwę użytkownika (login), uid, gid, prawdziwe imię i nazwisko konta, katalog domowy i powłokę logowania.

uid 0 – konta superużytkowników

# **Prawa dostępu**

Prawa dostępu do plików i katalogów są definiowane pod względem dostępu do **odczytu**, dostępu do **zapisu** oraz dostępu do **wykonania**

Pierwsze 10 znaków na liście to atrybuty pliku. Pierwszy z tych znaków to typ pliku. Pozostałe dziewięć znaków atrybutów pliku, nazywane **trybem pliku**

|  |  |  |
| --- | --- | --- |
| **Właściciel** | **Grupa** | **Świat (pozostali użytkownicy)** |
| rwx | rwx | rwx |

|  |  |  |
| --- | --- | --- |
| **Atrybut** | **Pliki** | **Katalogi** |
| r | Pozwala na otwarcie i odczyt pliku. | Pozwala na wylistowanie zawartości katalogu, jeśli ustawiony jest również atrybut wykonania. |
| w | Pozwala na zapisywanie do pliku lub jego skracanie; jednak ten atrybut nie pozwala na zmienianie nazwy ani usuwanie plików. Możliwość usuwania lub zmieniania nazwy plików jest określana przez atrybuty katalogu. | Pozwala na tworzenie, usuwanie i zmienianie nazwy plików w katalogu, jeśli ustawiony jest również atrybut wykonania. |
| x | Pozwala na potraktowanie pliku jako programu i jego wykonanie. Pliki programów napisane w językach skryptowych muszą być również ustawione jako możliwe do odczytu, aby mogły być wykonywane. | Pozwala na wejście do katalogu, np. za pomocą polecenia cd directory. |

chmod - służy do zmiany trybu (uprawnień) pliku lub katalogu (tylko właściciel może zmieniać)

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Ósemkowa** | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 |
| **Binarna** | 000 | 001 | 010 | 011 | 100 | 101 | 110 | 111 |
| **Tryb pliku** | --- | --x | -w- | -wx | r-- | r-x | rw- | rwx |

|  |  |
| --- | --- |
| **Symbol** | **Znaczenie** |
| u | Skrót od "user", ale oznacza właściciela pliku lub katalogu. |
| g | Właściciel grupy. |
| o | Skrót od "others", ale oznacza świat. |
| a | Skrót od "all". Jest to kombinacja u, g i o |

Jeśli nie zostanie określony żaden znak, zakłada się "all".

+ wskazującym, że uprawnienie ma być dodane - wskazującym, że uprawnienie ma być odebrane

= wskazującym, że mają być stosowane tylko określone uprawnienia, a wszystkie inne mają być usuwane.

**sudo** pozwala administratorowi skonfigurować plik o nazwie /etc/sudoers

**su** - służy do uruchomienia powłoki jako inny użytkownik

su - uruchamia powłokę dla superużytkownika

**su -c 'polecenie'** - wykonanie pojedynczego polecenia zamiast uruchamiania nowego interaktywnego polecenia

Ważną różnicą między su a sudo jest to, że sudo nie uruchamia nowej powłoki ani nie ładuje środowiska innego użytkownika. Oznacza to, że polecenia nie muszą być cytowane inaczej niż byłyby bez użycia sudo

sudo pyta o hasło użytkownia, nie superużytkownika

**sudo -i** - rozpoczęcia interaktywnej sesji superużytkownika **sudo -l** – wylistowane uprawnienia przez sudo

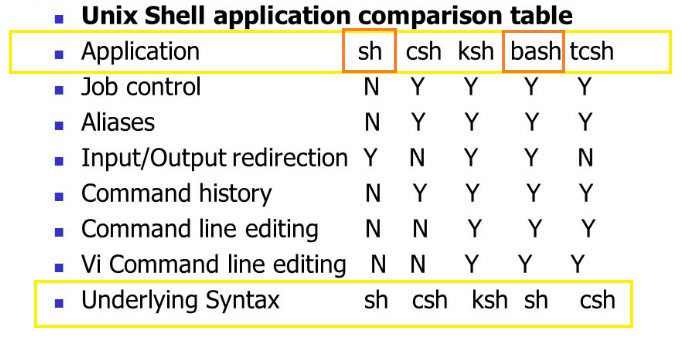
**chown** - służy do zmiany właściciela i właściciela grupy pliku lub katalogu. Do użycia tego polecenia wymagane są uprawnienia  
chown [owner][:[group]] file...

|  |  |
| --- | --- |
| **Argument** | **Wyniki** |
| bob | Zmienia właścicielstwo pliku z obecnego właściciela na użytkownika bob. |
| bob:users | Zmienia właścicielstwo pliku z obecnego właściciela na użytkownika bob i zmienia właściciela grupy pliku na grupę users. |
| :admins | Zmienia właściciela grupy na grupę admins. Właściciel pliku pozostaje niezmieniony. |
| bob: | Zmienia właściciela pliku z obecnego właściciela na użytkownika bob i zmienia właściciela grupy na grupę logowania użytkownika bob. |

chgrp to stare chown

passwd - ustawić lub zmienić hasło

# **bash**



Plik /etc/profile: Ten plik zawiera domyślne ustawienia systemowe dla wszystkich użytkowników korzystających z powłoki bash. Wykonywane są w nim polecenia zawarte w pliku, a także polecenia z plików .sh przechowywanych w katalogu /etc/profile.d.

/etc/profile.d: Katalog zawierający pliki skryptów shell z rozszerzeniem .sh, które są również uruchamiane podczas logowania. Pozwala to na elastyczną modyfikację ustawień bez bezpośredniej ingerencji w plik /etc/profile.

Pliki konfiguracyjne w katalogu domowym: Po wykonaniu pliku /etc/profile, bash poszukuje plików konfiguracyjnych w katalogu domowym użytkownika. Kolejność to ~/.bash\_profile, ~/.bash\_login, ~/.profile. Pierwszy napotkany plik zostanie wykonany, co pozwala na personalizację ustawień dla konkretnego użytkownika.

Pliki ~/.bashrc, /etc/bashrc: Pliki te są wywoływane, gdy zostanie uruchomiona nowa powłoka lub sesja. Są one odpowiedzialne za dodatkowe ustawienia i konfiguracje powłoki, zarówno globalne, jak i dla konkretnego użytkownika.

Plik ~/.bash\_logout: Zawiera polecenia, które są wykonywane podczas wylogowywania. Często używane do "sprzątania" przed zakończeniem sesji, na przykład usuwanie plików tymczasowych.

Eksportowanie zmiennych: Korzystanie z plików .sh w /etc/profile.d do ustawiania i eksportowania zmiennych zapewnia, że są one dostępne dla powłoki logowania każdego użytkownika.

Domyślne pliki dla nowych kont: W typowej dystrybucji systemu Linux nowe konta są tworzone z plikami ~/.bash\_profile oraz ~/.bashrc, które odpowiednio wywołują siebie nawzajem oraz plik /etc/bashrc.

Aktualizacje systemu: Korzystanie z plików w /etc/profile.d pozwala na zachowanie wprowadzonych zmian, nawet gdy plik /etc/profile zostanie zaktualizowany podczas aktualizacji systemu.

**Interaktywne powłoki typu non-login po uruchomieniu wykonują polecenia zapisane w pliku ~/.bashrc. Domyślnie plik ~/.bashrc wywołuje plik /etc/bashrc. Plik ten nie jest wywoływany przez powłokę bash — w wielu systemach Linux plik ten jest wywoływany przez plik ~/.bashrc**

Powłoki nieinteraktywne sprawdzają, czy istnieje zmienna środowiskowa o nazwie BASH\_ENV (lub ENV, jeżeli wykonywaną powłoką jest sh) i wykonują polecenia zapisane w pliku, którego nazwa jest przechowywana w tej zmiennej.

W pliku .bashrc polecenia są wykonywane wielokrotnie, a podpowłoki dziedziczą wszystkie zmienne, dlatego zaleca się umieszczanie poleceń modyfikujących zmienne w pliku .bash\_profile. Na przykład, dodanie ścieżki do zmiennej PATH powinno być umieszczone w .bash\_profile, co gwarantuje jednorazowe dodanie podczas logowania. Modyfikacje w .bash\_profile przenoszą się na podpowłoki, w przeciwieństwie do .bashrc, gdzie zmiany zastępują ustawienia dziedziczone.

Po edycji plików startowych, takich jak .bashrc, zmiany można wprowadzić bez ponownego logowania, używając poleceń source lub . (kropka). Te polecenia uruchamiają skrypt jako część bieżącego procesu, co natychmiast wprowadza zmiany w bieżącej sesji powłoki. Użycie source lub . umożliwia bezpośredni wpływ na powłokę, jednak należy być ostrożnym przy uruchamianiu dowolnych skryptów, aby uniknąć niepożądanych efektów. Standardowe uruchamianie skryptów bez source lub . ogranicza dostępność zmiennych do podpowłok wykonujących skrypt.

| **Symbol** | **Polecenie** |
| --- | --- |
| () | Podpowłoka |
| $() | Podstawienie polecenia |
| (()) | Obliczenia arytmetyczne; synonim polecenia let. |
| $(()) | Rozwijanie wyrażeń arytmetycznych |
| [] | Polecenie test. |
| [[]] | Wyrażenie warunkowe; podobne do [], ale dodaje porównywanie ciągów znaków. |

Skrypt powłoki to plik zawierający sekwencje poleceń, które mogą być wykonane przez powłokę systemu operacyjnego. Polecenia w skrypcie mogą obejmować różne operacje, takie jak uruchamianie narzędzi systemu Linux, programów czy innych skryptów. Skrypty mogą manipulować plikami, przekierowywać wejście/wyjście, oraz korzystać z poleceń kontrolujących przepływ sterowania. Działają one jak grupa poleceń, które modyfikują kolejność wykonywania poleceń w sposób podobny do struktur sterujących w językach programowania. Powłoka interpretuje i wykonuje polecenia ze skryptu, umożliwiając efektywne wykonywanie złożonych lub często powtarzających się zadań.

**Musimy mieć odpowiednie prawa do odczytu i wykonania pliku zawierającego skrypt. Często używa się polecenia chmod +x <nazwa\_skryptu>.**

**Jeżeli pierwsze dwa znaki skryptu to #! , system interpretuje pozostałe znaki tego wiersza jako bezwzględną ścieżkę do programu, który powinien wykonać dany skrypt. Taka ścieżka może wskazywać na dowolny program, a nie tylko na powłokę.**

|  |  |
| --- | --- |
| #!/bin/bash -e (errexit) | Powoduje zamknięcie powłoki, gdy działanie prostego polecenia zakończy się niepowodzeniem. |
| #!/bin/bash -u (nounset) | Powoduje wyświetlenie komunikatu błędu i zamknięcie powłoki, gdy program próbuje rozwinąć nieustawioną zmienną. |
| #!/bin/bash -x | Włącza tryb debugowania |
| # | Komentarze |

Polecenie bash , otworzy nową powłokę i bezpośrednio w niej wykona skrypt. Jednak wolniej. (lepiej chmod)

Jeżeli bash nie jest domyślną powłoką interaktywną lub jeżeli chcemy sprawdzić, jak dany skrypt działa z innymi powłokami, to możemy spróbować uruchomić go, podając jego nazwę jako argument polecenia, na przykład bash czy tcsh . Oryginalna powłoka Bourne Shell była wywoływana poleceniem sh . Choć obecnie możesz z poziomu powłoki bash wykonać polecenie sh , to jednak nie będzie to oryginalna powłoka Bourne Shell. Polecenie sh (/bin/sh) jest dowiązaniem symbolicznym do /bin/bash lub /bin/dash, czyli inaczej mówiąc, jest to po prostu inna nazwa dla polecenia bash lub dash . Kiedy wywołujemy powłokę bash przy użyciu polecenia sh , powłoka bash stara się dokładnie naśladować zachowanie powłoki Bourne Shell, aczkolwiek nie zawsze się to udaje.

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| (enter) | ; | & | | | |& | () | || -o | && -a | ;; |
| znak nowego wiersza | znak oddzielenia | zadanie działające w tle | potokowanie | standardowy potok błędów | grupowanie | OR | AND | przerwanie bloku |

kolejność & => d & e & f == f potem e i d

Nazwa zmiennej musi rozpoczynać się od litery lub znaku podkreślenia (musi zawierać tylko cyfry, litery lub \_)

|  |  |
| --- | --- |
| $0 | Nazwa samego skryptu |
| $1 | Argument pozycyjny; w tym wypadku pierwszy argument |
| ${10} | 10-ty argument; ponieważ liczba większa od 10, to należy to umieścić w nawiasach klamrowych |
| $# | Liczba argumentów wywołania skryptu |
| $\* | Reprezentuje wszystkie argumenty wywołania skryptu |

**Uwaga. Między nazwą zmiennej, a znakiem równości jak i wartości po niej nie może być znaku spacji.**

Zmienne PATH,HOME czy $USER, to przykłady zmiennych środowiskowych, które są definiowane przez samą powłokę. Aby wyświetlić zmienne środowiskowe należy użyć polecenia ***printenv***.

Istnieją dwa sposoby podstawiania wyników działania poleceń do zmiennej:

* Używanie grawisów (znaku `)
* Używanie zapisu ze znakiem dolara $( )

read -p ”Podaj imię: ” imie

Istnieje również możliwość wykonywania obliczeń za pomocą wbudowanego kalkulatora bc. Na przykład echo "12+5" | bc wypisze nam wartość 17. Tutaj nie mamy do czynienia ani ze słowem kluczowym let, ani z rozwijaniem wyrażeń arytmetycznych.

# **test, pętle, here**

test – zwraca 0 albo 1 (true, false)

test WYRAZENIE test ! WYRAZENIE - odwrócenie działania tego polecenia

Można użyć [ WYRAZENIE ]   
[ WYRAZENIE ] && echo "True" Jeżeli wyrażenie jest prawdziwe, to wyświetli się komunikat "True"

Test porównuje też ze sobą ciągi znaków   
test $ZMIENNA = zmienna / [ $zmienna = zmienna ]

Zmiennej zakończenia polecenia: $?

Sprawdza, czy ciąg znaków nie jest pusty  
test -n $SSH\_TTY / [ -n $SSH\_TTY ]

$SSH\_TTY jest specjalną zmienną, która sprawdza czy bieżące połączenie terminala jest wykonane przez SSH

Czy ciąg znaków ma zerową długość? Należy użyć opcji -z.  
test -z $1 / [ -z $1 ]

Może nie przechodzić dla niektórych powłok).

(( liczba1 > liczba2 )) #sprawdza czy liczba1 jest większa niż liczba2

| **Opcja** | **Opis** |
| --- | --- |
| -d | czy podany plik jest katalogiem |
| -e | czy plik istnieje |
| -x | czy plik jest wykonywalny |
| -f | czy plik jest zwykłym plikiem |
| -r | czy plik można odczytać |
| -p | czy plik jest nazwanym potokiem |
| -b | czy plik jest urządzeniem blokowym |
| -O | czy zalogowany użytkownik jest właścicielem pliku |
| -c | czy plik jest urządzeniem znakowym |
| plik1 -nt plik2 | sprawdza czy plik1 jest nowszy niż plik2 |
| plik1 -ot plik2 | sprawdza czy plik1 jest starszu niż plik2 |

|  |  |
| --- | --- |
| -eq | = |
| -ge | >= |
| -gt | > |
| -le | <= |
| -lt | < |
| -ne | != |

#!/bin/bash

if warunek; then

polecenie

polecenie

fi

if warunek; then

polecenie

polecenie

else

polecenie

polecenie

fi

Zamiast ‘;’ można dać nową linię

elif warunek; then

Case

case wyrazenie in

wartosc1)

polecenie

polecenie

;;

wartosc2)

polecenie

polecenie

;;

\*) <- jeśli nie znajdzie dopasowania

polecenie

polecenie

;;

esac

[[ wyrażenie ]] podobne do test  ale dodaje ważne nowe wyrażenie łańcuchowe

ciag1 =~ regex  
Powyższe wyrażenie zwraca prawdę, jeśli ciąg1 jest dopasowany do rozszerzonego wyrażenia regularnego regex

if [[ "$INT" =~ ^-?[0-9]+$ ]]; then sprawdza czy jest to liczba całkowita

[[ ]] – pozwala sprawdzić kolejność alfabetyczną

= umożliwia sprawdzenie czy ciąg znaków pasuje do podanego wzorca (wzorzec z PRAWEJ)

Program Here

Powłoka przesyła wszystko, co znajduje się pomiędzy tymi znacznikami,  na standardowe wejście procesu.   
Obraz zawierający tekst, Czcionka, zrzut ekranu

Opis wygenerowany automatycznie

while warunek

do

polecenie

polecenie

...

done

until

Aby przekierować plik do pętli, umieszczamy operator przekierowania po instrukcji done

for licznik\_petli in lista\_argumentow

do

polecenie

polecenie

...

done

# **Parametry**

|  |  |
| --- | --- |
| ${parameter:-word} | jeśli parameter nie jest ustawiony (tj. Nie istnieje) lub jest pusty, to interpretacja daje w wyniku wartość słowa. Jeśli parametr nie jest pusty, wynikiem rozwinięcia jest wartość parametru. |
| ${parameter:=word} | Jeśli parameter jest nieustawiony lub pusty, to interpretacja daje w wyniku wartość słowa. Dodatkowo do parameter przypisana jest wartość słowa. |
| ${parameter:?word} | Jeśli parameter jest nieustawiony lub pusty, to rozwinięcie powoduje zakończenie skryptu z błędem, a zawartość słowa jest wysyłana do standardowego błędu |
| ${parameter:+word} | jeśli parameter jest nieustawiony lub pusty, interpretacja nic nie daje. Jeśli parameter nie jest pusty, wartość słowa jest podstawiana za parameter; jednak wartość parameter nie ulega zmianie |

${!prefix\*}

${!prefix@}  
zwraca nazwy istniejących zmiennych, których nazwy rozpoczynają się od prefix

|  |  |
| --- | --- |
| ${#parameter} | rozwija się do długości łańcucha zawartego w parameter. jeśli parameter jest @ lub \*, to interpretacja daje w wyniku liczbę parametrów pozycyjnych |
| ${parameter:offset:length} | od początku ciągu i trwa do końca łańcucha, chyba że określono długość  Należy zwrócić uwagę, że wartości ujemne muszą być poprzedzone spacją, aby uniknąć pomylenia z rozszerzeniem ${parametr: -słowo} |
| ${parameter#pattern}  ${parameter##pattern} | pattern jest wzorcem wieloznacznym, takim jak używane przy rozwijaniu nazw plików. Różnica w tych dwóch formach polega na tym, że # usuwa najkrótsze dopasowanie, a formularz ## usuwa najdłuższe dopasowanie. |
| ${parameter%pattern}  ${parameter%%pattern} | Poniższe rozwinięcia są takie same jak poprzednie rozszerzenia # i ##, z tą różnicą, że usuwają tekst z końca ciągu zawartego w parametrze, a nie z początku. |

Jeśli zostanie znaleziony tekst pasujący do pattern , zostanie on zastąpiony zawartością string

|  |  |
| --- | --- |
| ${parameter/pattern/string} | pierwsze wystąpienie wzoru jest zastęp |
| ${parameter//pattern/string} | wszystkie wystąpienia są zastępowane |
| ${parameter/#pattern/string} | wymaga, aby dopasowanie wystąpiło na początku string |
| ${parameter/%pattern/string} | wymaga, aby dopasowanie wystąpiło na końcu string |

W każdej postaci można pominąć /string, powodując usunięcie tekstu dopasowanego do wzorca

typeset i declare

| **Atrybut** | **Znaczenie** |
| --- | --- |
| -a | Deklaruje zmienną jako tablicę |
| -f | Deklaruje zmienną jako nazwę funkcji |
| -i | Deklaruje zmienną jako wartość typu integer (całkowitego) |
| -r | Deklauje zmienną jako tylko do odczytu |
| -x | Powoduje wyeksportowanie zmiennej (nadaje jej status zmiennej globalnej) |
| -u | Traktuje wartość zmienną wielkimi literami |
| -l | Traktuje wartość zmiennej małymi literami |

declare -F wyświetla listę funkcji rezydujących w środowisku powłoki

declare -f wyświetla wszystkie funkcje wraz z ich definicjami

| **Operator** | **Opis** |
| --- | --- |
| ~ | Negacja bitowa. Zamienia 0 na 1, a 1 na 0 |
| << | Przesunięcie bitowe w lewo. Przesuwa wszystkie bity o daną liczbę w lewo |
| >> | Przesunięcie bitowe w prawo. Przesuwa wszystkie bity o daną liczbę w prawo |
| & | Bitowa koniunkcja. Wykonuje operację AND na każdym bicie dwóch liczb |
| | | Bitowa alternatywa. Wykonuje operację OR na każdym bicie dwóch liczb |
| ^ | Bitowa alternatywa wykluczająca. Wykonuje operację XOR na każdym bicie dwóch liczb |

(( $(echo "3.1415 > 30.1415" | bc -l) )) && echo "True"

# - ilość znaków lub ilość elementów

# **Markdown**

# (h1) ##(h2) nagłówek albo pod tekstem dodajemy dowolną liczbę znaków == dla nagłówka poziomu 1 lub -- znaków dla nagłówka poziomu 2  
# Nagłówek {identyfikator}  
[Link do nagłówka](#identyfikator)

[^1]: jęzek python

|  |  |
| --- | --- |
| </br> / 2spacje | nowa linia |
| \*tekst\* lub \_tekst\_ | kursywa |
| \*\*tekst\*\* lub \_\_tekst\_\_ | pogrubienie |
| \*\*\*tekst\*\*\* lub \_\_\_tekst\_\_\_ | pogrubienie i kursywa |
| ~~tekst~~ | przekreślenie |

>cytat  
>>cytat w cytacie itd.

![picture/git.jpg](picture/git.jpg) obraz z repo

Lista uporządkowana (zaczyna liczyć od 1 argumentu, jeśli pierwszy 12 to kolejny 13)

1. jeden
2. dwa
3. trzy

Listy nieuporządkowane (\*,+,-)

#Checklista

* [x ] Wstać
* [ ] zjeść

|nagłówek1| nagłówek2| nagłówek3|  
|:---|:---:|---:|  
|elem1| elem2| elem3|

`` lub ` do robienia fragment kodu

``` lub ~~~ do bloku kodu (można zrobić ```py do określenia języka

:wink: ==tekst== highlight

|  |  |
| --- | --- |
| **skrót** | **opis** |
| KDE | K Desktop Environment |
| CLI | Interfejs wiersza poleceń |
| GUI | Graficzny interfejs użytkownika |
| GNOME | GNU Network Object Model Environment |
| GNU | GNU's not Unix |

Polecenie może być jednym z czterech różnych rzeczy:

* Wykonywalnym programem
* Poleceniem wbudowanym w powłokę
* Funkcją powłoki
* Aliasem

Nawiasy kwadratowe -> elementy opcjonalne.   
Znak pionowej kreski -> elementy wzajemnie wykluczające się

cd [-L|[-P [-e]]] [dir] poleceniu cd może towarzyszyć opcjonalnie albo -L, albo -P, a jeśli opcja -P jest określona, opcja -e może być również uwzględniona, po której następuje opcjonalny argument dir

printenv | less - Aby zobaczyć listę dostępnych zmiennych, należy użyć tego polecenia

zadanie cron – zadanie uruchamiane okresowo

set | less Programowalne dopełnianie powłoki

# **Katalogi**

* /bin - Podstawowe pliki wykonywalne. Przechowuje pliki wykonywalne dostępne dla wszystkich użytkowników oraz pliki niezbędne do uruchomienia systemu w trybie single-user
* /boot - Statyczne pliki programu rozruchowego. Zawiera wszystkie pliki niezbędne do uruchomienia systemu.
* /dev - Pliki urządzeń. Zawiera pliki, które reprezentują urządzenia peryferyjne, takie jak dyski, terminale oraz drukarki.
* /etc - Pliki konfiguracyjne systemu lokalnego. Przechowuje pliki administracyjne, konfiguracyjne i inne pliki systemowe.
* /etc/opt - Pliki konfiguracyjne dla pakietów oprogramowania z katalogu /opt
* /etc/X11 - Pliki konfiguracyjne lokalnego serwera X Window System
* /home - Katalogi domowe użytkowników
* /lib - Współużytkowane biblioteki systemowe
* /mnt - Punkty montowania dla tymczasowo montowanych systemów plików
* /opt - Dodatkowe (opcjonalne) pakiety oprogramowania
* /root - Katalog domowy użytkownika root
* /sbin - Podstawowe, systemowe pliki wykonywalne (binarnia). Tutaj przechowywane są narzędzia używane przez administratorów systemów  oraz również narzędzia niezbędne podczas uruchamiania systemu.
* /tmp - Pliki tymczasowe
* /usr - Drugorzędowa hierarchia dla współużytkowanych danych użytkownika. Tradycyjnie znajdują się tutaj podkatalogi zawierające informację wykorzystywane przez system operacyjny.
* /usr/bin - Polecenia i narzędzia użytkowników. Znajdziesz tutaj standardowe programy narzędziowe systemu Linux (głównie binarnia, które nie są potrzebne w trybie single-user).
* /usr/include - Pliki nagłówkowe używane przez programy w języku C
* /usr/share - Dane niezależne od architektury systemu. Znajdziesz tam podkatalogi jakie jak locale, info, man.
* /var - Dane często ulegające zmianom. W tym katalogu przechowywane są pliki, których zawartość często się zmienia podczas działania systemu. Przykładami takich plików są pliki tymczasowe, pliki kolejek wydruku i skrzynki pocztowe użytkowników.

# **Teoria git**

System kontroli wersji (VCS)

Jednym z bardziej popularnych narzędzi VCS był system o nazwie RCS, który do dziś jest dystrybuowany na wielu komputerach. RCS działa poprzez przechowywanie zestawów poprawek w specjalnym formacie na dysku; może następnie odtworzyć, jak każdy plik wyglądał w dowolnym momencie, dodając wszystkie łatki.

Opracowano scentralizowane systemy kontroli wersji (CVCS). Systemy te, takie jak CVS, Subversion i Perforce, mają jeden serwer, który zawiera wszystkie wersjonowane pliki oraz pewną liczbę klientów, którzy pobierają pliki z tego centralnego miejsca.

Rozproszone systemy kontroli wersji (DVCS). W systemach DVCS (takich jak Git, Mercurial, Bazaar lub Darcs) klienci nie tylko sprawdzają najnowszą migawkę plików — w pełni odzwierciedlają repozytorium. Tak więc, jeśli któryś serwer umrze, a te systemy współpracują za jego pośrednictwem, dowolne z repozytoriów klientów można skopiować na serwer, aby go przywrócić. Każda migawka to tak naprawdę pełna kopia zapasowa wszystkich danych.

Narzędzie Git zaczęło się od odrobiny kreatywnej destrukcji i ognistych kontrowersji. Jądro Linuksa to projekt oprogramowania typu open source o dość dużym zakresie. Przez większość czasu utrzymania jądra Linuksa (1991–2002) zmiany w oprogramowaniu były przekazywane jako łatki i pliki archiwalne. W 2002 roku projekt jądra Linux zaczął używać zastrzeżonego DVCS o nazwie BitKeeper. W 2005 roku zerwały się relacje między społecznością, która opracowała jądro Linuksa, a komercyjną firmą, która opracowała BitKeepera, a status darmowego narzędzia został cofnięty. To skłoniło społeczność programistów Linuksa (w szczególności Linusa Torvaldsa, twórcę Linuksa) do opracowania własnego narzędzia w oparciu o wnioski, których nauczyli się podczas korzystania z BitKeepera. Do cech nowego systemu miała należeć szybkość, prostota, wsparcie równoległości, rozproszenie oraz obsługa dużych projektów. Od momentu powstania w 2005 roku, Git ewoluował i dojrzał, aby był łatwy w użyciu, a jednocześnie zachował te początkowe cechy. Jest niesamowicie szybki, bardzo wydajny przy dużych projektach i ma niesamowity system rozgałęzień do nieliniowego rozwoju. Jest stosowany nie tylko w tworzeniu oprogramowania, ale też w pracach redakcyjnych itd.

(CVS, Subversion, Perforce, Bazaar itd.) traktują przechowywane przez siebie informacje jako zestaw plików i zmiany wprowadzane do każdego pliku w czasie. Git nie myśli ani nie przechowuje swoich danych w ten sposób. Zamiast tego Git myśli o swoich danych bardziej jak o zestawie migawek miniaturowego systemu plików. Za każdym razem, gdy zatwierdzane lub zapisywany jest stan projektu w Git, w zasadzie robi on obraz tego, jak w danym momencie wyglądają wszystkie pliki i przechowuje odniesienie do tego zrzutu. Aby być wydajnym, jeśli pliki się nie zmieniły, Git nie przechowuje go ponownie, tylko link do poprzedniego identycznego pliku, który już zapisał. Git myśli o swoich danych bardziej jak o strumieniu migawek.

Wszystko w Git jest sumowane przed zapisaniem, a następnie jest określane przez tzw. sumę kontrolną. Oznacza to, że nie można zmienić zawartości dowolnego pliku lub katalogu bez powiadomienia o tym narzędzia Git. Ta funkcjonalność jest wbudowana w Git na najniższych poziomach i jest integralną częścią jego filozofii. Nie możemy utracić informacji podczas przesyłania ani uzyskać uszkodzenia pliku bez możliwości jego wykrycia przez Git. Mechanizm używany przez Git do tej sumy kontrolnej nazywa się skrótem SHA-1. Jest to 40-znakowy ciąg składający się ze znaków szesnastkowych (0–9 i a–f) i obliczany na podstawie zawartości pliku lub struktury katalogów w Git. Skrót SHA-1 wygląda mniej więcej tak:

24b9da6552252987aa493b52f8696cd6d3b00373

Git ma trzy główne stany, w których mogą znajdować się pliki: zatwierdzone (commited), zmodyfikowane (modified) i umieszczone w poczekalni (staged) lub inaczej przemieszczony.

Pierwszą rzeczą, którą powinniśmy zrobić podczas instalacji Git, jest ustawienie nazwy użytkownika i adresu e-mail. Jest to ważne, ponieważ każde zatwierdzenie Git wykorzystuje te informacje i są one nieodwołalnie zapisywane w zatwierdzeniach, które zaczynamy tworzyć:

Kiedy zauważymy projekt, który nam się podoba, możemy śledzić go jak w mediach społecznościowych i obserwować jego postępy. Jeśli chcemy popracować nad nową funkcją lub naprawić błąd, wystarczy zrobić klon projektu i nad nim pracować. Ten proces nazywa się „Forking”

Po wprowadzeniu wszystkich zmian w kopii projektu możesz przesłać Pull Request (PR) do opiekuna projektu.

GitHub Pages to po prostu witryna internetowa hostowana bezpośrednio w repozytorium

tag lekki - zachowuje się mniej więcej tak jak gałąź, która się nie zmienia - jest to tylko wskaźnik do konkretnej rewizji

tag opisany (tag -a) - są przechowywane jako pełne obiekty w bazie danych Git. Są one opatrywane: sumą kontrolną, nazwiskiem, adres e-mail, data. Posiadają notkę etykiety, mogą być podpisywane i weryfikowane za pomocą GNU Privacy Guard (GPG).

Jeżeli dodamy kilka zmian i zatwierdzimy je ponownie, kolejny commit będzie zawierał wskaźnik do commita poprzedniego

**date** -  służy do wyświetlania i ustawiania w systemie odpowiedniej daty i czasu systemowego

| **Parametr** | **Opis parametru** |
| --- | --- |
| %H | Wyświetlenie obecnej godziny w formacie 24-godzinnym |
| %I | Wyświetlenie aktualnej godziny w formacie 12-godzinnym |
| %M | Wyświetlanie aktualnej minuty, bez godziny |
| %p | Sprawdzenie, czy czas podany w postaci 12-godzinnej jest przed południem i po południu |
| %r | Wyświetlanie czasu w formacie 12-godzinnym |
| %s | Wskazanie ile sekund minęło od 01.01.1970 |
| %S | Wyświetlenie liczby sekund, które upłynęły od pełnej minuty |
| %T | Wyświetlenie pełnego czasu godzin wraz z minutami i sekundami |
| +%X | Wyświetlenie lokalnej reprezentacji na komputerze |
| %b | Wyświetlenie skróconej nazwy obecnego miesiąca |
| %B | Wyświetlenie pełnej lokalnej nazwy miesiąca |
| %c | Wyświetlenie lokalnej daty i czasu w miejscu, w którym się znajdujemy |
| %d | Wyświetlenie aktualnego dnia miesiąca |
| %D | Wyświetlenie daty w formacie miesiąc/dzień/rok |
| %j | Wyświetlenie obecnego dnia roku |
| %m | Wyświetlenie informacji o obecnym numerze miesiąca |
| %U | Wyświetlenie informacji o obecnym numerze tygodnia |
| %w | Wyświetlenie numeru dnia tygodnia, gdzie 0 oznacza niedzielę |
| %W | Wyświetlenie numeru tygodnia, który rozpoczyna się od poniedziałku |
| %x | Wyświetlenie lokalną reprezentację daty w formacie RRRR-MM-DD |
| %y | Wyświetlenie dwóch ostatnich cyfr roku |
| %Y | Wyświetlenie pełnego roku w notacji czterocyfrowej |
| -u | Wyświetlenie daty w czasie UTC |
| -r | Wyświetlenie daty modyfikacji pliku |
| -R | Wyświetlenie daty w formacie RFC 5322 |
| -d, --date=string | Wyświetlenie daty podanego jako string |

# **Skróty klawiszowe**

|  |  |
| --- | --- |
| ctrl-A | Przesuń kursor na początek linii. |
| ctrl-E | Przesuń kursor na koniec linii. |
| ctrl-F | Przesuń kursor o jeden znak do przodu; tak samo jak klawisz strzałki w prawo. |
| ctrl-B | Przesuń kursor o jeden znak do tyłu; tak samo jak klawisz strzałki w lewo. |
| alt-F | Przesuń kursor o jedno słowo do przodu. |
| alt-B | Przesuń kursor o jedno słowo do tyłu. |
| ctrl-L | Wyczyść ekran i przesuń kursor do lewego górnego rogu. Polecenie clear robi to samo. |

skróty używane do edycji znaków w linii poleceń

|  |  |
| --- | --- |
| ctrl-D | Usuń znak w miejscu kursora. |
| ctrl-T | Zamień miejscami znak w miejscu kursora z poprzedzającym go znakiem. |
| alt-T | Zamień miejscami słowo w miejscu kursora z poprzedzającym go słowem. |
| alt-L | Zamień znaki od miejsca kursora do końca słowa na małe litery. |
| alt-U | Zamień znaki od miejsca kursora do końca słowa na wielkie litery. |

Dokumentacja Readline używa terminów killing i yanking (wycinanie, wklejanie). Elementy, które są wycinane, są przechowywane w buforze (tymczasowym obszarze pamięci) nazywanym kill-ring

|  |  |
| --- | --- |
| ctrl-K | Wycięcie tekstu od miejsca kursora do końca linii. |
| ctrl-U | Wycięcie tekstu od miejsca kursora do początku linii. |
| alt-D | Wycięcie tekstu od miejsca kursora do końca bieżącego słowa. |
| alt- backspace | Wycięcie tekstu od miejsca kursora do początku bieżącego słowa. Jeśli kursor znajduje się na początku słowa, wycina poprzednie słowo. |
| ctrl-Y | Wklej tekst z kill-ring i wstaw go w miejscu kursora. |

|  |  |
| --- | --- |
| alt-? | Wyświetl listę możliwych dopełnień. Na większości systemów można to również zrobić, naciskając klawisz tab drugi raz, co jest znacznie łatwiejsze. |
| alt-\* | Wstaw wszystkie możliwe dopełnienia. Jest to przydatne, gdy chcesz użyć więcej niż jednego możliwego dopasowania. |