

Wprowadzenie do systemu Unix

Wszystkie komendy z ćwiczeń z opisem działania

Spis treści:

- Wymagania do 1. Kolokwium, opis podstawowych komend - **2**
- Pliki /etc/passwd, /etc/group, /etc/shadow, /etc/gshadow - **4**
 - Historia poleceń, wprowadzenie - **6**
- Wypisywanie jednej linii z pliku: komendy head i tail - **7**
 - Manual - **8**
- Dwa najpopularniejsze interpretery w /etc/passwd: komendy cat oraz cut - **10**
 - Polecenie awk - **11**
- Wyszukiwanie plików katalogu po znakach - **13**
 - Tworzenie i usuwanie katalogów - **15**
 - Operacje na plikach - **16**
 - Prawa dostępu do plików - **18**
 - Prawa dostępu do katalogów - **20**
 - Umaska - **22**
- Prawa dostępu do katalogu domowego - **23**
 - Systemy plików - **24**
- Polecenia z galaxy oraz shell-lab - **25**
- Wyszukiwanie plików: komenda find - **26**
 - Procesy - **27**

Przygotowanie do 1. Kolokwium:

Wymagania:

Obowiązujące komendy

- man
- cat, more, less
- head, tail
- who, last, w, id
- cal
- pwd
- echo
- ls
- wc
- awk
- uniq, tr, cut, sort
- grep
- find
- ps, jobs
- kill
- df, du
- lsblk
- stat

Obowiązujące pliki i katalogi

- /etc/passwd, /etc/group, /etc/shadow, /etc/gshadow
- /etc/fstab
- /, /var, /usr

Obowiązująca składnia

- | (pipe)
- przekierowania: , >>, 2>, 2>>
- wyrażenia regularne (zgodne z grep)
- glob (globbing) • sygnały (kill -l)
- Ctrl-c, Ctrl-z
- command substitution (\${ ... })

Przykładowe zadanie • Podaj PID procesu, który zużywa najwięcej pamięci rezydującej spośród procesów, których rodzicem jest proces o PID równym 1?

.....

w- Udziela informacji o użytkownikach aktualnie zalogowanych i wykonywanych przez nich programach.

who- Udziela informacji o zalogowanych użytkownikach

last- Udostępnia informację o historii logowań użytkowników

echo - Wyświetlanie tekstu ze standardowego wejścia (klawiatura). Za pomocą polecenia *echo*, możemy wyświetlić w terminalu tekst, który wpisujemy z klawiatury. Również za pomocą polecenia *echo* możemy przenieść tekst do pliku.

tr - Przetłumacz, albo usuń znaki. Polecenie *tr* umożliwia zmianę znaków na inne, usuwanie nadmiernej ilości znaków (np. spacji, kropek, itp.).

W linuxie masz do dyspozycji komendę **kill**, której zadaniem jest wysłanie sygnału do procesu, w celu jego zamknięcia.

df- Wyświetlanie informacji o wolnym miejscu na dysku. Polecenie *df* daje możliwość sprawdzenia w jakim stopniu mamy zajęty dysk twardy.

du- Ile miejsca na dysku zajmuje plik (katalog). Aby sprawdzić ile miejsca na dysku zajmuje plik (katalog), można użyć polecenia *du*.

lsblk- to polecenie listujące urządzenia blokowe

stat- Informacje o pliku, lub systemie plików

Gdy potrzebujemy dowiedzieć się paru informacji o pliku, a do dyspozycji mamy jedynie konsolę, możemy użyć polecenia **stat**, które to wyświetli przydatne informacje o pliku. Na rys. 1. pokazano jakie informacje zostają wyświetlane. W skład informacji wchodzi: Nazwa pliku, rozmiar, uprawnienia, UID oraz GID właściciela pliku. Informacja o

rozdaju pliku (tutaj plik zwykły - regular file). Ilość dowiązań (links: 1). Data ostatniego dostępu oraz modyfikacji pliku.

```
gruby@arch-tux ~/Desktop/test$ stat plik1.txt
  File: `plik1.txt'
  Size: 438          Blocks: 8          IO Block: 4096   regular file
Device: 801h/2049d  Inode: 1591406      Links: 1
Access: (0644/-rw-r--r--)  Uid: ( 1000/   gruby)   Gid: ( 1000/   gruby)
Access: 2011-12-14 00:10:52.890757829 +0100
Modify: 2011-12-14 00:09:39.327711394 +0100
Change: 2011-12-14 00:09:39.327711394 +0100
 Birth: -
gruby@arch-tux ~/Desktop/test$
```

Podstawowy plik konfiguracyjny (/etc/passwd)

- Plik tekstowy
- Zawartość może odczytać każdy użytkownik
- Informacja zapisana jest w 7 -ciu kolumnach oddzielonych znakiem „:”
- Kolumny:
 - 1) Nazwa użytkownika (login name)
 - 2) Dawniej zakodowana postać hasła, obecnie niektóre systemy przechowują informacje o stanie hasła
 - 3) Numer identyfikacyjny użytkownika (UID)
 - 4) Numer grupy podstawowej użytkownika (GID)
 - 5) Opis użytkownika
 - 6) Bezwzględna ścieżka dostępu do katalogu domowego użytkownika (HOME)
 - 7) Bezwzględna ścieżka dostępu do podstawowego interpretera użytkownika (SHELL)

/etc/group

- Plik tekstowy
- Jedna linia opisuje jedną grupę
- Każda linia podzielona jest na 4 pola oddzielone znakiem „:”
- Kolumny:
 - 1) Nazwa grupy (unikalna w systemie)
 - 2) Obecnie informacja o stanie hasła
 - 3) Numer grupy w systemie
 - 4) Lista członków grupy

/etc/shadow

- Plik tekstowy
- Budowa linikowa – jedna linijka-jeden użytkownik
- Każda linijka ma 9 pól oddzielonych „:”
- Prawa dostępu umożliwiają edycję tylko użytkownikowi root
- Kolumny:
 - 1) Nazwa użytkownika w systemie (login name)
 - 2) Postać zakodowana hasła
 - 3) Liczba dni, licząc od 1 stycznia 1970r., kiedy hasło było ostatni raz zmienione
 - 4) Liczba dni przed upływem których zmiana hasła nie jest możliwa
 - 5) Liczba dni po upływie których konieczna jest zmiana hasła
 - 6) Liczba dni, jaka dzieli hasło od przedawnienia, kiedy użytkownik będzie o tym fakcie powiadomiony.
 - 7) Liczba dni, po wygaśnięciu hasła, kiedy konto jest blokowane (interwencja administratora)
 - 8) Liczba dni, licząc od 1 stycznia 1970r., po upływie których konto jest blokowane. Wykorzystywane przy zakładaniu kont tymczasowych.
 - 9) Kolumna 9 – pole zarezerwowane

/etc/gshadow

- Przechowuje hasło grupowe oraz informacje o administratorach grupy
- Kolumny:
 - 1) kolumna 1 – nazwa grupy w systemie
 - 2) kolumna 2 – hasło grupowe
 - 3) kolumna 3 – lista administratorów grupy
 - 4) kolumna 4 – lista członków grupy

Historia Poleceń

1. Wprowadzenie:

- pwd

pwd - Wyświetlenie ścieżki dostępu do bieżącego katalogu. Za pomocą polecenia pwd możemy wyświetlić informację, gdzie się znajdujemy. To znaczy ścieżkę do katalogu, w którym aktualnie się znajdujemy.

- cat /etc/passwd

cat - Wyświetlanie zawartości pliku na standardowe wyjście. Do obejrzenia zawartości pliku (zawartość wyświetlana w terminalu), bez konieczności otwierania pliku w edytorze tekstowym służy polecenie cat, wraz z różnymi parametrami ułatwia nawigację po wyświetlanych danych.

- file /etc/passwd
- file /bin/lis
- file /usr/include/stdio.h

file - do rozpoznawania zawartości pliku.

- more /etc/passwd

Polecenia more, less będą pomocne, kiedy zechcemy przeczytać plik lub kiedy rezultat wykonania jakiegoś polecenia nie zmieści się na ekranie. Polecenie more pozwala nam poruszać się tylko w dół pliku.

- less /etc/passwd
- less /usr/include/stdio.h

Polecenie less jest podobne do more, ale umożliwia poruszanie się w przód i w tył pliku lub wyniku polecenia.

- wc /etc/passwd
- wc -l /etc/passwd
- wc -w /etc/passwd
- wc -c /etc/passwd
- wc -wc /etc/passwd
- wc -cw /etc/passwd

Drukowanie informacji o zawartości pliku. Za pomocą polecenia wc można dowiedzieć się (nie otwierając pliku w żadnym edytorze tekstowym) informacji na temat ilości bajtów, słów, znaków, linii, oraz rozmiarze najdłuższej linii.

wc /etc/passwd:

Zlicza liczbę linii, słów i bajtów w pliku /etc/passwd i wyświetla te informacje.

wc -l /etc/passwd:

Zlicza tylko liczby linii w pliku /etc/passwd i wyświetla tę informację.

wc -w /etc/passwd:

Zlicza tylko liczby słów w pliku /etc/passwd i wyświetla tę informację.

wc -c /etc/passwd:

Zlicza tylko liczbę bajtów w pliku /etc/passwd i wyświetla tę informację.

wc -wc /etc/passwd:

Nie jest poprawnym użyciem opcji. wc oczekuje jednej opcji, a tutaj użyto dwóch (-wc). W rezultacie może wystąpić błąd lub wyświetlone zostaną informacje o błędzie.

wc -cw /etc/passwd:

Opcje -cw są równoważne i można je podać w dowolnej kolejności. Zlicza liczbę słów i bajtów w pliku /etc/passwd i wyświetla te informacje.

- cal
- cal 23
- cal 9 1752

Narzędzie **cal** wyświetla prosty kalendarz w tradycyjnym formacie

- bc
- bc -l

Polecenie bc jest używane do kalkulatora wiersza poleceń. Jest podobny do podstawowego kalkulatora, za pomocą którego możemy wykonać podstawowe obliczenia matematyczne.

- head /etc/passwd
- head -3 /etc/passwd

Wyświetlanie początkowej części pliku. Jeśli potrzebujemy wyświetlić tylko początek pliku (np. plik .c, aby dowiedzieć się jakie dołączone są biblioteki), to nie musimy od razu wyświetlać całego pliku, tylko możemy posłużyć się poleceniem head.

- tail /etc/passwd
- tail -1 /etc/passwd

tail - wyświetlenie ostatnich części plików. Polecenie tail służy do wyświetlania ostatnich wierszy (lub, bajtów) danego pliku.

Wypisywanie jednej linii z pliku

- head -17 /etc/passwd
- head -17 /etc/passwd | tail -1
- cat /etc/passwd | head -17
- cat /etc/passwd | tail -n+17 | head -1

head -17 /etc/passwd

Wyświetla pierwsze 17 linii pliku /etc/passwd.

head -17 /etc/passwd | tail -1

Wyświetla 17. linię pliku /etc/passwd za pomocą połączenia komend head i tail. head -17 wybiera pierwsze 17 linii, a tail -1 wybiera ostatnią z tych linii.

cat /etc/passwd | head -17

Wyświetla pierwsze 17 linii pliku /etc/passwd. W tym przypadku korzysta z komendy cat do wyświetlenia całego pliku, a następnie head ogranicza wyniki do pierwszych 17 linii.

cat /etc/passwd | tail -n+17 | head -1

Wyświetla 17. linię pliku /etc/passwd bez używania head -17. tail -n+17 wybiera linie od 17. w górę, a head -1 wybiera pierwszą z tych linii.

Manual

- man grep
- man passwd
- man 5 passwd

man - Podręcznik systemowy. Bardzo często zdarza się tak (ośmielim się powiedzieć, że codziennie), że nie znamy dokładnej składni polecenia, bądź nie wiemy, jakie argumenty można przekazać do polecenia, w tej sytuacji z pomocą zdecydowanie przychodzi podręcznik systemowy, w którym jest pełna dokumentacja danego programu.

Wyszukiwanie linii w pliku /usr/include/stdio.h

- grep FILE /usr/include/stdio.h
- grep -i FILE /usr/include/stdio.h
- grep -i FILE /usr/include/stdio.h | wc -l
- grep -ic FILE /usr/include/stdio.h
- grep -ivc FILE /usr/include/stdio.h
- wc -l /usr/include/stdio.h
- grep -iw /usr/include/stdio.h

grep FILE /usr/include/stdio.h

Wyszukuje linie w pliku /usr/include/stdio.h, które zawierają ciąg znaków "FILE".

grep -i FILE /usr/include/stdio.h

Wyszukuje linie w pliku /usr/include/stdio.h, ignorując wielkość liter, dla ciągu znaków "FILE".

grep -i FILE /usr/include/stdio.h | wc -l

Wyszukuje linie w pliku /usr/include/stdio.h dla ciągu znaków "FILE", ignorując wielkość liter, a następnie używa wc -l do zliczenia liczby znalezionych linii.

grep -ic FILE /usr/include/stdio.h

To samo co poprzednia komenda, ale zamiast wc -l, używa opcji -c aby zliczyć liczbę wystąpień "FILE".

grep -ivc FILE /usr/include/stdio.

Wyszukuje linie, które nie zawierają ciągu znaków "FILE", ignorując wielkość liter, a następnie zlicza ilość tych linii.

wc -l /usr/include/stdio.h

Zlicza liczby linii w pliku /usr/include/stdio.h.

grep -iw /usr/include/stdio.h

Wyszukuje linie w pliku /usr/include/stdio.h dla dokładnie pasującego ciągu znaków "FILE" (ignorując wielkość liter).

Narzędzie służące do wyszukiwania.

Najczęściej używane opcje:

- **-i** ignoruje wielkość liter
- **-r** wyszukuje również w sub katalogach ścieżki
- **-l** pokazuje nazwy plików, w których znajduje się pasujący wzorzec
- **-n** pokazuje linię, w której występuje szukany wzorzec
- **-v** pokazuje linie, które nie zawierają wzorca
- **-F** traktuje szukany wzorzec, jako escapowany string
- **-E** traktuje szukany wzorzec, jako rozszerzone wyrażenie regularne
- **-e** informuje, że następny argument, jest wzorcem
- **-c** zwróci nam liczbę pasujących fraz
- **-x** zwróci wynik, tylko i wyłącznie, jeżeli pasuje cała linia
- **-m[cyfra]** przerwie wyszukiwanie, po znalezieniu pierwszych [cyfra] rezultatów

Opis użytkownika o UID 14

- **cat /etc/passwd | grep 14**
- **cat /etc/passwd | grep 1**
- **cat /etc/passwd | cut -c1,4,7-9**
- **cat /etc/passwd | cut -d: -f3,5**
- **cat /etc/passwd | cut -d: -f3,5 | grep 14**
- **cat /etc/passwd | cut -d: -f3,5 | grep '^14:'**
- **cat /etc/passwd | cut -d: -f3,5 | grep '^14:' | cut -d: -f2**

cat /etc/passwd | grep 14:

Wyświetla linie z pliku /etc/passwd, w których występuje liczba 14.

cat /etc/passwd | grep 1:

Wyświetla linie z pliku /etc/passwd, w których występuje cyfra 1.

cat /etc/passwd | cut -c1,4,7-9:

Wyświetla określone kolumny (znaki) z pliku /etc/passwd. Kolumny to: 1, 4, oraz 7-9.

cat /etc/passwd | cut -d: -f3,5:

Używa cut do wycięcia pól z pliku /etc/passwd, oddzielonych dwukropkiem (:). Wybiera trzecie i piąte pole, które to zazwyczaj zawierają UID i pełne imię i nazwisko użytkownika.

cat /etc/passwd | cut -d: -f3,5 | grep 14:

Wycina trzecie i piąte pole z pliku /etc/passwd (UID i pełne imię i nazwisko użytkownika) i następnie filtruje tylko te linie, w których występuje liczba 14.

cat /etc/passwd | cut -d: -f3,5 | grep '^14:':

Podobnie jak poprzednia komenda, ale dodatkowo używa grep z ^14: aby dopasować linie, które zaczynają się od 14:. To zazwyczaj odpowiada określonemu UID.

cat /etc/passwd | cut -d: -f3,5 | grep '^14:' | cut -d: -f2:

Kombinuje wcześniejsze komendy. Wycina drugie pole (imię i nazwisko użytkownika) z linii, które pasują do UID rozpoczynającego się od 14:.

Dwa najpopularniejsze interpretery w /etc/passwd

- man 5 passwd
- cat /etc/passwd | cut -d: -f7
- cat /etc/passwd | cut -d: -f7 | sort
- cat /etc/passwd | cut -d: -f7 | sort | uniq -c
- cat /etc/passwd | cut -d: -f7 | sort | uniq -c | sort -r
- cat /etc/passwd | cut -d: -f7 | sort | uniq -c | sort -rn
- cat /etc/passwd | cut -d: -f7 | sort | uniq -c | sort -rn | head -2

Komenda **cut** umożliwia wycięcie bajtów, znaków lub pól z każdego wiersza pliku i zapisanie tych bajtów, znaków lub pól na wyjściu standardowym. Jeśli parametr *Plik* nie zostanie określony, komenda **cut** odczytuje standardowe wejście.

man 5 passwd:

Wyświetla podręcznik (manpage) dla pliku /etc/passwd z sekcji 5. Sekcja 5 w manualu Linuxa dotyczy plików konfiguracyjnych i formatów plików.

cat /etc/passwd | cut -d: -f7:

Wyświetla siódme pole z pliku /etc/passwd dla każdej linii, które jest zazwyczaj informacją o powłoce (shell) użytkownika.

cat /etc/passwd | cut -d: -f7 | sort:

Wyświetla siódme pole z pliku /etc/passwd dla każdej linii, sortuje je alfabetycznie.

```
cat /etc/passwd | cut -d: -f7 | sort | uniq -c:
```

Wyświetla siódme pole z pliku /etc/passwd dla każdej linii, sortuje je, a następnie używa uniq -c do zliczenia liczby unikalnych wartości i wyświetlenia ich.

```
cat /etc/passwd | cut -d: -f7 | sort | uniq -c | sort -r:
```

Wyświetla siódme pole z pliku /etc/passwd dla każdej linii, sortuje je, a następnie używa uniq -c do zliczenia liczby unikalnych wartości. Następnie sortuje te wartości w odwrotnej kolejności według liczby wystąpień.

```
cat /etc/passwd | cut -d: -f7 | sort | uniq -c | sort -rn:
```

To samo co poprzednia komenda, ale używa skróconego zapisu -rn zamiast -r -n do sortowania w odwrotnej kolejności według liczby wystąpień.

```
cat /etc/passwd | cut -d: -f7 | sort | uniq -c | sort -rn | head -2:
```

Wyświetla siódme pole z pliku /etc/passwd dla każdej linii, sortuje je, a następnie używa uniq -c do zliczenia liczby unikalnych wartości. Następnie sortuje te wartości w odwrotnej kolejności według liczby wystąpień, a na końcu używa head -2, aby ograniczyć wyniki do dwóch pierwszych linii (najczęściej występujących powłok).

Polecenie uniq w systemie UNIX to narzędzie wiersza poleceń do raportowania lub filtrowania powtarzających się wierszy w pliku. Jest w stanie usunąć duplikaty, wyświetlić sumę wystąpień, wyświetlić tylko powtórzone wiersze, zignorować poszczególne znaki i porównać na podstawie określonych zmiennych.

Polecenie awk - Służy do wyszukiwania wierszy w plikach, które są zgodne ze wzorcem i wykonuje określone działania na tych wierszach

- cat /etc/passwd | awk -F: '{ print \$5 }'
- cat /etc/passwd | awk -F: '{ if (\$3 == 14) print \$5 }'
- cat /etc/passwd | awk -F: '{ if (\$5 == "") print \$1 }'
- cat /etc/passwd | awk -F: '{ if (\$5 == "") print \$1 \$3 }'
- cat /etc/passwd | awk -F: '{ if (\$5 == "") print \$1 " "\$3 }'
- cat /etc/passwd | awk -F: '{ if (\$5 == "") print \$1 " -> "\$3 }'
- cat /etc/passwd | awk -F: '{ if (\$5 == "") print \$3 " "\$1 }'

```
➤ cat /etc/passwd | awk -F: '{ print $5 }'
```

- Wyświetla piąte pole z pliku /etc/passwd dla każdej linii, używając dwukropka (:) jako separatora pól.

```
➤ cat /etc/passwd | awk -F: '{ if ( $3 == 14 ) print $5 }'
```

- Wyświetla piąte pole z pliku /etc/passwd tylko dla tych linii, gdzie trzecie pole (UID) wynosi 14.

```
➤ cat /etc/passwd | awk -F: '{ if ( $5 == "" ) print $1 }'
```

- Wyświetla pierwsze pole (nazwę użytkownika) z pliku `/etc/passwd` dla tych linii, gdzie piąte pole (pełna nazwa użytkownika) jest puste.
- `cat /etc/passwd | awk -F: '{ if ($5 == "") print $1 $3 }'`
- Wyświetla połączone pierwsze pole (nazwę użytkownika) i trzecie pole (UID) z pliku `/etc/passwd` dla tych linii, gdzie piąte pole (pełna nazwa użytkownika) jest puste.
- `cat /etc/passwd | awk -F: '{ if ($5 == "") print $1 " "$3 }'`
- To samo co poprzednia komenda, ale używa spacji jako separatora między pierwszym polem (nazwą użytkownika) a trzecim polem (UID).
- `cat /etc/passwd | awk -F: '{ if ($5 == "") print $1 " -> "$3 }'`
- Wyświetla połączone pierwsze pole (nazwę użytkownika) i trzecie pole (UID) z pliku `/etc/passwd` dla tych linii, gdzie piąte pole (pełna nazwa użytkownika) jest puste, z dodaną strzałką.
- `cat /etc/passwd | awk -F: '{ if ($5 == "") print $3 " "$1 }'`
- Wyświetla połączone trzecie pole (UID) i pierwsze pole (nazwę użytkownika) z pliku `/etc/passwd` dla tych linii, gdzie piąte pole (pełna nazwa użytkownika) jest puste.

Awk - Służy do wyszukiwania wierszy w plikach, które są zgodne ze wzorcem i wykonuje określone działania na tych wierszach.

Dowolny ciąg znaków i jeden znak w nazwach plików i wyrażeniu regularnym (wyszukiwanie plików w katalogu po znakach)

- `ls /usr/include/st*.h`
- `ls /usr/include/s?????`
- `ls /usr/include/s???????`
- `ls /usr`
- `ls -d /usr`
- `ls /usr/include/s???????`
- `ls -d /usr/include/s???????`
- `ls /usr/include/`
- `ls /usr/include/ | less`
- `ls /usr/include/ | grep st*.h`
- `ls /usr/include/ | grep '^st*.h'`
- `ls /usr/include/ | grep '^st.*.h'`
- `ls /usr/include/ | grep '^st.*\..h'`

- `ls /usr/include/ | grep '^st.*\.h$'`
- `ls /usr/include/ | grep '^s.....'`
- `ls /usr/include/ | grep '^s.....$'`
- `ls /usr/include/ | grep '^s.\{6\}$'`
- `ls /usr/include/ | grep '^s.\{4,6\}$'`
- `ls /usr/include/ | grep '^s.\{4,\}$'`
- `ls /usr/include/ | grep '^s.\{,6\}$'`

ls- Wyświetlenie zawartości katalogu. W konsoli, aby wyświetlić zawartość katalogu trzeba użyć jednego z kilku dostępnych poleceń, nie mniej jednak polecenie `ls` jest chyba najbardziej popularnym.

1. `ls /usr/include/st*.h`

- Wyświetla pliki w katalogu `/usr/include/` zaczynające się od "st" i kończące się na ".h".

2. `ls /usr/include/s?????`

- Wyświetla pliki w katalogu `/usr/include/` zaczynające się od "s" i mające dokładnie pięć dowolnych znaków.

3. `ls /usr/include/s??????`

- Wyświetla pliki w katalogu `/usr/include/` zaczynające się od "s" i mające dokładnie sześć dowolnych znaków.

4. `ls /usr`

- Wyświetla zawartość katalogu `/usr`.

5. `ls -d /usr`

- Wyświetla informacje o katalogu `/usr`, bez wchodzenia do jego wnętrza.

6. `ls /usr/include/s??????`

- Wyświetla pliki w katalogu `/usr/include/` zaczynające się od "s" i mające dokładnie sześć dowolnych znaków.

7. `ls -d /usr/include/s??????`

- Wyświetla informacje o katalogu `/usr/include/`, bez wchodzenia do jego wnętrza, dla plików zaczynających się od "s" i mających dokładnie sześć dowolnych znaków.

8. `ls /usr/include/`

- Wyświetla zawartość katalogu `/usr/include/`.

9. `ls /usr/include/ | less`

- Wyświetla zawartość katalogu `/usr/include/` strona po stronie za pomocą programu `less`.

10. `ls /usr/include/ | grep st*.h`

- Wyświetla linie zawierające ciąg "st*.h" w nazwach plików w katalogu `/usr/include/`. Jednak `*` w `grep` oznacza dowolny ciąg znaków, więc może nie działać oczekiwanie.

11. `ls /usr/include/ | grep '^st*.h'`

- Próbuje dopasować linie, które zaczynają się od "st*.h". Jednak `*` w `grep` działa inaczej niż w poleceniach obsługi powłoki, co może prowadzić do nieoczekiwanych wyników. Poprawne użycie `.` dla dowolnego ciągu znaków.

12. `ls /usr/include/ | grep '^st.*.h'`

- Wyświetla linie zawierające ciąg "st" i dowolny ciąg znaków, a następnie ".h" w katalogu `/usr/include/`.

13. `ls /usr/include/ | grep '^st.*\\.h'`

- Wyświetla linie zawierające ciąg "st" i dowolny ciąg znaków, a następnie ".h" w katalogu `/usr/include/`. Teraz `\\` oznacza dosłownie kropkę.

14. `ls /usr/include/ | grep '^st.*\\.h$'`

- Wyświetla linie zawierające ciąg "st" i dowolny ciąg znaków, a następnie ".h", które są jednocześnie końcem linii w katalogu `/usr/include/`.

15. `ls /usr/include/ | grep '^s.....'`

- Wyświetla linie zawierające ciąg "s....." w katalogu `/usr/include/`. `.....` oznacza dokładnie sześć dowolnych znaków.

16. `ls /usr/include/ | grep '^s.....$'`

- Wyświetla linie zawierające ciąg "s.....", które są jednocześnie końcem linii w katalogu `/usr/include/`.

17. `ls /usr/include/ | grep '^s.{6}$'`

- Wyświetla linie zawierające ciąg "s" i dokładnie sześć dowolnych znaków, które są jednocześnie końcem linii w katalogu `/usr/include/`. Używa `{6}` do określenia ilości powtórzeń.

18. `ls /usr/include/ | grep '^s.{4,6}$'`

- Wyświetla linie zawierające ciąg "s" i od czterech do sześciu dowolnych znaków, które są jednocześnie końcem linii w katalogu `/usr/include/`. Używa `{4,6}` do określenia zakresu ilości powtórzeń.

19. `ls /usr/include/ | grep '^s.{4,}$'`

- Wyświetla linie zawierające ciąg "s" i co najmniej cztery dowolne znaki, które są jednocześnie końcem linii w katalogu `/usr/include/`. Używa `\{4,\}` do określenia minimalnej ilości powtórzeń.

20. `ls /usr/include/ | grep '^s.\{6\}$'`

- Wyświetla linie zawierające ciąg "s" i co najwyżej sześć dowolnych znaków, które są jednocześnie końcem linii w katalogu `/usr/include/`. Używa `\{,6\}` do określenia maksymalnej ilości powtórzeń.

Tworzenie i usuwanie katalogów:

- `cd`
- `mkdir c3/txt c3/bin`
- `mkdir c3 c3/txt c3/bin`
- `ls -liR c3`
- `rmdir c3`
- `rm -rf c3`
- `ls -liR c3`
- `mkdir -p c3/txt c3/bin`
- `ls -liR c3`
- `cd c3/txt`
- `pwd`

cd- Zmiana katalogu. W konsoli, aby zmienić katalog, trzeba użyć polecenia `cd`.

21. `cd`:

- Przechodzi do katalogu domowego bieżącego użytkownika.

22. `mkdir c3/txt c3/bin`:

- Tworzy katalogi `c3`, `c3/txt` i `c3/bin` w bieżącym katalogu.

23. `mkdir c3 c3/txt c3/bin`:

- To samo co poprzednia komenda, ale używa jednej komendy do utworzenia wszystkich trzech katalogów.

24. `ls -liR c3`:

- Wyświetla rekurencyjnie (`-R`) zawartość katalogu `c3`, wraz z informacjami o plikach (`-l`) i numerami i-uznaniami (`-i`).

25. `rmdir c3`:

- Usuwa katalog `c3` tylko wtedy, gdy jest pusty.
26. `rm -rf c3`:
- Usuwa katalog `c3` oraz jego zawartość (opcja `-r` oznacza rekurencyjne usuwanie, a `-f` oznacza wymuszenie bez pytania o potwierdzenie).
27. `ls -alR c3`:
- Sprawdza zawartość katalogu `c3` po usunięciu. Katalog nie istnieje, więc nie ma żadnej zawartości do wyświetlenia.
28. `mkdir -p c3/txt c3/bin`:
- Tworzy katalogi `c3`, `c3/txt` i `c3/bin`, ale używa opcji `-p`, dzięki czemu nie generuje błędów, jeśli katalogi już istnieją.
29. `ls -alR c3`:
- Wyświetla rekurencyjnie zawartość katalogu `c3`, wraz z informacjami o plikach i numerami i-uznaniami.
30. `cd c3/txt`:
- Przechodzi do katalogu `c3/txt`.
31. `pwd`:
- Wyświetla bieżący katalog roboczy (`/ścieżka/do/katalogu/c3/txt`).

Warto zauważyć, że komendy usuwające (`rmdir` i `rm -rf`) są potencjalnie niebezpieczne, ponieważ skutkują utratą danych. Zawsze należy uważać podczas korzystania z nich, aby nie utracić ważnych informacji.

Operacje na plikach:

- `man ls > ls.txt`
- `ls -l`
- `file ls.txt`
- `less ls.txt`
- `cp ls.txt ls.bak`
- `ls -l`
- `diff ls.txt ls.bak`
- `cp /bin/tar`
- `ls -l`
- `cp /bin/find /bin/ls /etc/passwd`
- `ls -l`
- `mv ls mls`

cp - Kopiowanie plików (katalogów). Aby kopiować pliki lub katalogi za pomocą konsoli musimy użyć polecenia **cp**. Polecenia tego można użyć w trzech wariantach **diff** - oznacza różnicę . To polecenie służy do wyświetlania różnic w plikach poprzez porównywanie plików linia po linii

mv - Przenieś, lub zmień nazwę pliku / katalogu. Aby przenieść pliki lub katalogi za pomocą konsoli używa się polecenia **mv**. Za pomocą tego polecenia również zmienia się nazwy plików, lub katalogów.

1. `man ls > ls.txt`

- Tworzy plik o nazwie `ls.txt` i zapisuje w nim dokumentację dla polecenia `ls` za pomocą komendy `man`.

2. `ls -l`

- Wyświetla zawartość bieżącego katalogu, w tym pliki i katalogi, wraz z ich szczegółowymi informacjami (opcja `-l`).

3. `file ls.txt`

- Wyświetla informacje o typie pliku dla `ls.txt`.

4. `less ls.txt`

- Przegląda zawartość pliku `ls.txt` za pomocą programu `less`.

5. `cp ls.txt ls.bak`

- Kopiuje zawartość pliku `ls.txt` do nowego pliku o nazwie `ls.bak`.

6. `ls -l`

- Wyświetla zawartość bieżącego katalogu, w tym nowo utworzony plik `ls.bak`.

7. `diff ls.txt ls.bak`

- Porównuje różnice między plikami `ls.txt` i `ls.bak`.

8. `cp /bin/tar .`

- Kopiuje program `tar` z katalogu `/bin` do bieżącego katalogu (`.`).

9. `ls -l`

- Wyświetla zawartość bieżącego katalogu, w tym nowo skopiowany plik `tar`.

10. `cp /bin/find /bin/ls /etc/passwd .`

- Kopiuje programy `find` i `ls` z katalogu `/bin` oraz plik `/etc/passwd` do bieżącego katalogu (`.`).

11. `ls -l`

- Wyświetla zawartość bieżącego katalogu, w tym nowo skopiowane pliki `find`, `ls`, i `passwd`.

12. `mv ls mls`

- Zmienia nazwę pliku `ls` na `mls` za pomocą polecenia `mv`.

Prawa dostępu do plików:

- `ls -l`
- `id`
- `tail ls.txt`
- `chmod u-r ls.txt`
- `ls -l`
- `tail ls.txt`
- `date`
- `date >> ls.txt`
- `ls -l`
- `chmod u+r,u-w ls.txt`
- `ls -l`
- `tail ls.txt`
- `diff ls.txt ls.bak`
- `diff ls.bak ls.txt`
- `date >> ls.txt`
- `rm ls.txt`
- `ls -l`
- `yes`
- `./ls.bak`
- `chmod a+x ls.bak`
- `ls -l`
- `./ls.bak`
- `head ls.bak`

id - Wyświetlanie prawdziwego i efektywnego ID. Polecenia `id` używa się do wyświetlenia informacji o użytkowniku i grupie, podanej jako argument, lub jeśli argument nie występuje to o aktualnie zalogowanym.

ls - Zmiana dostępu do plików (prawa dostępu). W Linuksie każdy plik (katalog) ma swoje określone prawa dostępu. Definiują one czy plik może zostać odczytany, czy można do niego pisać, czy można go wykonać. Prawa dostępu mogą występować w różnych kombinacjach. Aby sprawdzić prawa dostępu, trzeba użyć polecenia `ls` z opcją `-l`.

rm - Usuwanie plików / katalogów. Aby usuwać pliki lub katalogi, lub jednocześnie to i to za pomocą konsoli, należy użyć polecenia `rm`. W zależności od tego, czy są to tylko pliki, puste katalogi, czy katalogi z zawartością, trzeba użyć odpowiedniej opcji w poleceniu.

1. `ls -l`:

- Wyświetla listę plików w bieżącym katalogu, wraz z informacjami dotyczącymi praw dostępu.

2. `id`:

- Wyświetla informacje o bieżącym użytkowniku, grupie itp.

3. `tail ls.txt:`

- Wyświetla ostatnie linie pliku `ls.txt`.

4. `chmod u-r ls.txt:`

- Usuwa uprawnienie do odczytu dla właściciela pliku `ls.txt`.

5. `ls -l:`

- Wyświetla listę plików w bieżącym katalogu po zmianie uprawnień pliku `ls.txt`.

6. `tail ls.txt:`

- Próbuje wyświetlić zawartość pliku `ls.txt`. Prawdopodobnie zostanie wyświetlony komunikat o braku dostępu.

7. `date:`

- Wyświetla aktualną datę i czas.

8. `date >> ls.txt:`

- Dodaje aktualną datę i czas na końcu pliku `ls.txt`.

9. `ls -l:`

- Wyświetla listę plików w bieżącym katalogu po dodaniu daty do pliku `ls.txt`.

10. `chmod u+r,u-w ls.txt:`

- Dodaje uprawnienie do odczytu i usuwa uprawnienie do zapisu dla właściciela pliku `ls.txt`.

11. `ls -l:`

- Wyświetla listę plików w bieżącym katalogu po ponownej zmianie uprawnień pliku `ls.txt`.

12. `tail ls.txt:`

- Wyświetla ostatnie linie pliku `ls.txt`.

13. `diff ls.txt ls.bak:`

- Porównuje różnice między plikami `ls.txt` i `ls.bak`.

14. `diff ls.bak ls.txt:`

- Porównuje różnice między plikami `ls.bak` i `ls.txt`.

15. `date >> ls.txt:`

- Dodaje aktualną datę i czas na końcu pliku `ls.txt`.

16. `rm ls.txt:`

- Usuwa plik `ls.txt`.

17. `ls -l:`

- Wyświetla listę plików w bieżącym katalogu po usunięciu pliku `ls.txt`.

18. `yes:`

- Continuously generates and prints the string "y" (potentially used for an interactive script).

19. `./ls.bak:`

- Wykonuje plik wykonywalny `ls.bak`.

20. `chmod a+x ls.bak`:

- Dodaje uprawnienia do wykonania dla wszystkich (użytkowników, grupy, innych) do pliku `ls.bak`.

21. `ls -l`:

- Wyświetla listę plików w bieżącym katalogu po dodaniu uprawnień do wykonania do pliku `ls.bak`.

22. `./ls.bak`:

- Ponownie wykonuje plik `ls.bak` po dodaniu uprawnień do wykonania.

23. `head ls.bak`:

- Wyświetla początkowe linie pliku `ls.bak`.

Prawa dostępu do katalogów:

- `ls -al`
- `chmod u-r .`
- `ls -al`
- `touch plik1`
- `rm mls`
- `chmod u-w,u+r .`
- `ls -al`
- `rm passwd`
- `touch plik2`
- `chmod u+w,u-x .`
- `ls`
- `cd ..`
- `ls -al`
- `cd txt`
- `ls txt`
- `ls -al txt`
- `ls -al`
- `chmod 702 txt`
- `ls -al`
- `chmod 5736 txt`
- Polecenie **touch** pozwala na utworzenie nowego pustego pliku w wierszu poleceń systemu Linux.
- Na przykład, wpisz **touch /twojadomena.pl/public_html/strona.html**, aby utworzyć plik HTML o nazwie **strona** w katalogu domowym domeny **twojadomena.pl**.

➤ `ls -al:`

- Wyświetla listę plików i katalogów wraz z informacjami dotyczącymi praw dostępu.

➤ `chmod u-r :`

- Usuwa uprawnienie do odczytu dla właściciela bieżącego katalogu (`.`).

➤ `ls -al:`

- Wyświetla listę plików i katalogów po usunięciu uprawnienia do odczytu dla katalogu.

➤ `touch plik1:`

- Tworzy pusty plik o nazwie `plik1`.

➤ `rm mls:`

- Usuwa katalog `mls`.

➤ `chmod u-w,u+r :`

- Usuwa uprawnienie do zapisu i dodaje uprawnienie do odczytu dla właściciela katalogu (`.`).

➤ `ls -al:`

- Wyświetla listę plików i katalogów po zmianie uprawnień dla katalogu.

➤ `rm passwd:`

- Usuwa plik o nazwie `passwd`.

➤ `touch plik2:`

- Tworzy pusty plik o nazwie `plik2`.

➤ `chmod u+w,u-x :`

- Dodaje uprawnienie do zapisu i usuwa uprawnienie do wykonania dla właściciela katalogu (`.`).

➤ `ls:`

- Wyświetla zawartość bieżącego katalogu.

➤ `cd ..:`

- Przechodzi do katalogu nadrzędnego.

➤ `ls -al:`

- Wyświetla listę plików i katalogów wraz z informacjami dotyczącymi praw dostępu.

➤ `cd txt:`

- Przechodzi do katalogu `txt`.

➤ `ls txt:`

- Wyświetla zawartość katalogu `txt`.

➤ `ls -al txt:`

- Wyświetla informacje o prawach dostępu dla plików i katalogów w katalogu `txt`.

➤ `ls -al:`

- Wyświetla informacje o prawach dostępu dla plików i katalogów w bieżącym katalogu.

➤ `chmod 702 txt:`

- Nadaje uprawnienia 7 (czytanie, zapisywanie, wykonanie) dla właściciela, uprawnienia 0 dla grupy i uprawnienia 2 (tylko wykonanie) dla innych dla katalogu `txt`.

➤ `ls -al`:

- Wyświetla informacje o prawach dostępu po zmianie uprawnień dla katalogu `txt`.

➤ `chmod 5736 txt`:

- Nadaje uprawnienia 5 (czytanie, wykonanie) dla właściciela, uprawnienia 7 (czytanie, zapisywanie, wykonanie) dla grupy i uprawnienia 3 (czytanie, wykonanie) dla innych dla katalogu `txt`.

Umaska:

- `ls -al`
- `cd bin`
- `touch p1`
- `mkdir k1`
- `ls -l`
- `umask`
- `umask 0027`
- `mkdir k2`
- `touch p2`
- `ls -l`
- `cd`
- `umask 0002`
- `ls -l`

mkdir - Tworzenie katalogu. Aby utworzyć katalog z użyciem konsoli musimy użyć polecenia `mkdir`. Polecenie to bez żadnych dodatkowych opcji tworzy katalog podany jako argument. Jeśli argumentów będzie więcej, to odpowiednia liczba katalogów zostanie utworzona.

Maska użytkownika (ang. user mask)

Polecenie **umask** jest odpowiedzialne za ograniczenie praw dostępu (**chmod**). W jaki sposób to działa? A więc na początku zastanówmy się nad jedną rzeczą, a mianowicie podczas tworzenia pliku (katalogu) nadawane są jakieś uprawnienia dla tego pliku (katalogu). No dobrze, to jakie są te uprawnienia? Tutaj właśnie wchodzi rola polecenia `umask`. Standardowo w systemie Linux uprawnienia dla tworzonego pliku są `rw-rw-rw-` (ósemkowo 666) jeśli maska jest wyłączona. Z kolei jeśli tworzymy katalog i maska również jest wyłączona, to katalog posiadać będzie uprawnienia `rw-rw-rw-` (ósemkowo

777). Sprawdzenie wartości maski realizuje się używając polecenia `umask`, bez dodatkowych argumentów.

1. <code>ls -al</code> :	<ul style="list-style-type: none">Wyświetla listę plików i katalogów wraz z informacjami dotyczącymi praw dostępu.
2. <code>cd bin</code> :	<ul style="list-style-type: none">Przechodzi do katalogu <code>bin</code>.
3. <code>touch p1</code> :	<ul style="list-style-type: none">Tworzy pusty plik o nazwie <code>p1</code>.
4. <code>mkdir k1</code> :	<ul style="list-style-type: none">Tworzy katalog o nazwie <code>k1</code>.
5. <code>ls -l</code> :	<ul style="list-style-type: none">Wyświetla informacje o prawach dostępu dla plików i katalogów w bieżącym katalogu.
6. <code>umask</code> :	<ul style="list-style-type: none">Wyświetla aktualne ustawienia umaski.
7. <code>umask 0027</code> :	<ul style="list-style-type: none">Ustawia umaskę na 0027.
8. <code>mkdir k2</code> :	<ul style="list-style-type: none">Tworzy katalog o nazwie <code>k2</code> z uwzględnieniem aktualnej umaski.
9. <code>touch p2</code> :	<ul style="list-style-type: none">Tworzy pusty plik o nazwie <code>p2</code> z uwzględnieniem aktualnej umaski.
10. <code>ls -l</code> :	<ul style="list-style-type: none">Wyświetla informacje o prawach dostępu dla plików i katalogów w bieżącym katalogu po zmianie umaski.
11. <code>cd</code> :	<ul style="list-style-type: none">Przechodzi do katalogu domowego.
12. <code>umask 0002</code> :	<ul style="list-style-type: none">Ustawia umaskę na 0002.
13. <code>ls -l</code> :	<ul style="list-style-type: none">Wyświetla informacje o prawach dostępu dla plików i katalogów w katalogu domowym po zmianie umaski.

Prawa dostępu do katalogu domowego

- `ls -ld .`
- `pwd`
- `ls -ld $HOME`
- `chmod 700 $HOME`

➤ **cd**

➤ **pwd**

➤ **ls -ld :**

- Wyświetla szczegółowe informacje o bieżącym katalogu (`.`).
- `-d`: Pokazuje informacje o katalogu, a nie jego zawartości.

➤ **pwd:**

- Wyświetla pełną ścieżkę bieżącego katalogu.

➤ **ls -ld \$HOME:**

- Wyświetla szczegółowe informacje o katalogu domowym użytkownika.
- `$HOME`: Zmienna środowiskowa reprezentująca katalog domowy.

➤ **chmod 700 \$HOME:**

- Nadaje właścicielowi pełne uprawnienia do odczytu, zapisu i wykonania, a innym (grupie i pozostałym użytkownikom) nie nadaje żadnych uprawnień do katalogu domowego.
- `$HOME`: Zmienna środowiskowa reprezentująca katalog domowy.

➤ **cd:**

- Przechodzi do katalogu domowego.

➤ **pwd:**

- Wyświetla pełną ścieżkę bieżącego katalogu (którym jest teraz katalog domowy).

Systemy plików:

➤ **df**

➤ **df -i**

➤ **df:**

- Wyświetla informacje o dostępnym miejscu na dyskach.
- Pokazuje ilość użytej i dostępnej przestrzeni na dyskach podmontowanych w systemie plików.
- Domyślnie wyświetla informacje w kilobajtach.

➤ **df -i:**

- Wyświetla informacje o używanych i dostępnych elementach indeksu (inodach) na dyskach.
- Pokazuje ilość użytych i dostępnych inodów dla każdego systemu plików.
- Elementy indeksu to metadane przechowujące informacje o plikach (np. nazwa, typ, uprawnienia).

Z galaxy:

➤ quota -v

- `quota`: Komenda do zarządzania i wyświetlania limitów dyskowych.
- `-v`: Opcja "verbose" lub "szczegółowa", która powoduje wyświetlanie bardziej szczegółowych informacji.

Z shell-lab

- `du`
- `du -s /usr/*`
- `du -s /usr/* 2>/dev/null`
- `du -s /usr/* 2>/dev/null | sort -nr | head -1`

du:

Komenda du służy do wyświetlania informacji o użyciu przestrzeni dyskowej w danym katalogu i jego podkatalogach. Domyślnie pokazuje informacje w kilobajtach.

du -s /usr/*:

Wyświetla informacje o użyciu przestrzeni dyskowej w podkatalogach /usr/.

-s: Podsumowuje użycie przestrzeni dyskowej w każdym podkatalogu.

/usr/*: Oznacza wszystkie podkatalogi w katalogu /usr.

du -s /usr/* 2>/dev/null:

Dodaje 2>/dev/null, aby przekierować komunikaty o błędach (stderr) do /dev/null. To eliminuje ewentualne błędy związane z brakiem dostępu do niektórych katalogów.

du -s /usr/* 2>/dev/null | sort -nr | head -1:

Dodaje przekierowanie do sortowania i wyświetla tylko największy katalog.

sort -nr: Sortuje wyniki malejąco według rozmiaru.

head -1: Wyświetla tylko pierwszą linię, czyli największy katalog.

Wyszukiwanie plików:

- `find /usr/include -name "st*.h" -print`
- `find /usr/include -name "st*" -print`
- `find /usr/include -name "st*" -exec ls -l {} \;`
- `find /usr/include -name "st*" -exec ls -ld {} \;`
- `find /usr/include -name "s*" -type d -exec ls -ld {} \;`
- `find /usr -perm /6000 -exec ls -ld {} \;`

Znajdź pliki zaczynające się od "st" i kończące się na ".h" w /usr/include

```
find /usr/include -name "st*.h" -print
```

Znajdź pliki i katalogi zaczynające się od "st" w /usr/include

```
find /usr/include -name "st*" -print
```

Wykonaj `ls -l` dla plików i katalogów zaczynających się od "st" w /usr/include

```
find /usr/include -name "st*" -exec ls -l {} \;
```

Wykonaj `ls -ld` dla plików i katalogów zaczynających się od "st" w /usr/include

```
find /usr/include -name "st*" -exec ls -ld {} \;
```

Wykonaj `ls -ld` dla katalogów zaczynających się od "s" w /usr/include

```
find /usr/include -name "s*" -type d -exec ls -ld {} \;
```

Znajdź pliki i katalogi z ustawionym bitem setuid, setgid lub sticky bit w /usr

```
find /usr -perm /6000 -exec ls -ld {} \;
```

Procesy:

- `cd`

- `mkdir c4`
- `rm -rf c4`
- `mkdir c4`
- `cd c4`
- `wget galaxy.agh.edu.pl/~boryczko/prog.c`
- `wget galaxy.agh.edu.pl/~boryczko/signal.c`
- `file prog.c`
- `cat prog.c`
- `gcc -oprogram prog.c`
- `ls -l`
- `file prog`
- `gcc -o signal signal.c`

Oczywiście, przejdźmy przez każdą komendę krok po kroku:

1. ``cd``: Zmiana katalogu. Ta komenda służy do zmiany bieżącego katalogu roboczego.
2. ``mkdir c4``: Utworzenie nowego katalogu o nazwie "c4".
3. ``rm -rf c4``: Usunięcie katalogu "c4" i jego zawartości siłą (``-r`` dla rekurencyjnego, ``-f`` dla wymuszenia).
4. ``mkdir c4``: Ponowne utworzenie nowego katalogu o nazwie "c4".
5. ``cd c4``: Zmiana bieżącego katalogu roboczego na "c4".
6. ``wget galaxy.agh.edu.pl/~boryczko/prog.c``: Pobranie pliku "prog.c" z podanego adresu URL.
7. ``wget galaxy.agh.edu.pl/~boryczko/signal.c``: Pobranie pliku "signal.c" z podanego adresu URL.

8. ``file prog.c``: Określenie i wyświetlenie typu pliku "prog.c".
9. ``cat prog.c``: Wyświetlenie zawartości pliku "prog.c".
10. ``gcc -o prog prog.c``: Kompilacja programu C z pliku "prog.c" i utworzenie pliku wykonywalnego o nazwie "prog".
11. ``ls -l``: Wyświetlenie listy plików w bieżącym katalogu w formacie rozszerzonym.
12. ``file prog``: Określenie i wyświetlenie typu pliku "prog".
13. ``gcc -o signal signal.c``: Kompilacja programu C z pliku "signal.c" i utworzenie pliku wykonywalnego o nazwie "signal".
14. ``./prog``: Uruchomienie programu "prog".
15. ``jobs``: Wyświetlenie listy zadań w tle.
16. ``bg %1``: Przesunięcie zadania o identyfikatorze 1 do tła.
17. ``jobs``: Ponowne wyświetlenie listy zadań w tle.
18. ``fg %1``: Przywrócenie zadania o identyfikatorze 1 do pierwszego planu.
19. ``jobs``: Wyświetlenie listy zadań w tle.

20. ``fg %2``: Przywrócenie zadania o identyfikatorze 2 do pierwszego planu.

21. ``jobs``: Wyświetlenie listy zadań w tle.

22. ``bg``: Przesunięcie bieżącego zadania do tła.

23. ``jobs``: Wyświetlenie listy zadań w tle.

24. ``fg``: Przywrócenie najnowszego zadania w tle do pierwszego planu.

25. ``./prog &``: Uruchomienie programu "prog" w tle.

26. ``jobs``: Wyświetlenie listy zadań w tle.

27. ``jobs``: Ponowne wyświetlenie listy zadań w tle.

28. ``ps -l``: Wyświetlenie migawkowego stanu bieżących procesów w formacie rozszerzonym.

29. ``ps -le``: Wyświetlenie migawkowego stanu wszystkich procesów w formacie rozszerzonym.

30. ``ps -le | tail -n+2 | awk '{print $12}'``: Wyświetlenie 12. kolumny (polecenia) wszystkich procesów, pomijając nagłówki.

31. ``ps -le | tail -n+2 | awk '{print $12}' | grep -c '^?$``: Zliczenie procesów z pustym poleceniem.

32. ``ps au``: Wyświetlenie migawkowego stanu wszystkich procesów w formacie zorientowanym na użytkownika.

33. ``ps aux | tail -n+2 | awk '{print $6" "$1}'``: Wyświetlenie użycia pamięci i nazwy użytkownika wszystkich procesów (bez nagłówka).

34. ``ps aux | tail -n+2 | awk '{print $6" "$1}' | sort -nr``: Sortowanie procesów według użycia pamięci w kolejności malejącej.

35. ``ps aux | tail -n+2 | awk '{print $6" "$1}' | sort -nr | head -1``: Wyświetlenie procesu z najwyższym użyciem pamięci.

36. ``ps aux | tail -n+2 | awk '{print $6" "$1}' | sort -nr | head -1 | awk '{print $2}'``: Wyświetlenie identyfikatora PID procesu o najwyższym użyciu pamięci.

37. ``ps au``: Wyświetlenie migawkowego stanu wszystkich procesów w formacie zorientowanym na użytkownika.

38. ``kill -l``: Wyświetlenie listy wszystkich dostępnych sygnałów.

39. ``kill -STOP 3396383``: Wstrzymanie (zatrzymanie) procesu o identyfikatorze PID 3396383.

40. ``ps au``: Wyświetlenie migawkowego stanu wszystkich procesów.

41. ``kill -CONT 3396383``: Kontynuacja (wznowienie) procesu o identyfikatorze PID 3396383.

42. ``ps au``: Wyświetlenie migawkowego stanu wszystkich procesów.

43. ``kill -STOP 3396383``: Wstrzymanie (zatrzymanie) procesu o identyfikatorze PID 3396383.

44. ``ps au``: Wyświetlenie migawkowego stanu wszystkich procesów.

45. ``kill 3396383``: Zakończenie (zabicie) procesu o identyfikatorze PID 3396383.

46. ``ps au``: Wyświetlenie migawkowego stanu wszystkich procesów.

47. ``kill -CONT 3396383``: Kontynuacja (wznowienie) procesu o identyfikatorze PID 3396383.

48. ``ps au``: Wyświetlenie migawkowego stanu wszystkich procesów.

49. ``.`./prog &``: Uruchomienie programu "prog" w tle

50. ``kill -STOP 3403004``: Wstrzymanie (zatrzymanie) procesu o identyfikatorze PID 3403004.

51. ``ps au``: Wyświetlenie migawkowego stanu wszystkich procesów.

52. ``kill -9 3403004``: Przymusowe zakończenie (zabicie) procesu o identyfikatorze PID 3403004.

53. ``ps au``: Wyświetlenie migawkowego stanu wszystkich procesów.