

The background of the slide features a large, light blue watermark of the Poznań University of Technology logo. The logo is circular and contains a central shield with a crown on top, two figures holding a shield, and a castle tower. The text 'POLITECHNIKA POZNAŃSKA' is written along the top arc, and 'POZNAN UNIVERSITY OF TECHNOLOGY' is written along the bottom arc.

Institute of Computing Science
Poznan University of Technology

Systemy Operacyjne

podstawy pracy w systemie linux

Andrzej Stroiński
andrzej.stroinski@cs.put.edu.pl

February 28, 2017



Wstęp

- Materiały
- Zasady zaliczenia
- Motywacja

Środowisko pracy

- terminal
- Polecenia

Pomoc systemowa

- help
- man
- apropos
- info

Interpreter poleceń

- history
- skrót terminala
- nazwy uogólnione



- ▶ Warto przejrzeć inne źródła:
 - ▶ Użytkowanie systemu operacyjnego UNIX - *dr D.Wawrzyniak*
 - ▶ Systemy operacyjne - skrypt - *dr C.Sobaniec*
 - ▶ Strona przedmiotu
 - ▶ Strona *Dr A.Kobusińskiej*
 - ▶ Strona *K.Sieka*
 - ▶ pomoc systemowa `man`
 - ▶ inne...



Na ocenę końcową z laboratorium składają się:

- ▶ test — 60% oceny końcowej
- ▶ skrypt — 30% oceny końcowej
- ▶ inne (wejściówki, zadania domowe, aktywność) — 10% oceny końcowej

Wymagana jest obecność na wszystkich zajęciach

- ▶ na dwóch zajęciach wolno nie być bez podania przyczyny — nie wpływa to na ocenę końcową
- ▶ brak obecności na połowie lub więcej zajęć skutkuje niezaliczeniem zajęć



Dlaczego linux:

- ▶ darmowy
- ▶ open source
- ▶ wiodące rozwiązanie serwerowe

Środowisko:

- ▶ tekstowe
- ▶ graficzne (Unity, KDE, GNOME, MATE, Pantheon)

Dystrybucje:

- ▶ Debian, Ubuntu, Mint, ElementaryOS (.deb)
- ▶ OpenSuse, Fedora (.rpm)
- ▶ Arch, Manjaro (.pact)

- ▶ Zaloguj się do systemu podając login i hasło.
- ▶ Uruchom program Terminal (pozycja *Terminal Program* z menu)

Terminal

To interfejs pomiędzy użytkownikiem a jądrem systemu operacyjnego. W momencie zalogowania się użytkownika, program "logujący" sprawdza czy zgadzają się podane login i hasło. W przypadku udanego logowania uruchamiany jest program powłoki. Powłoka jest interpreterem linii poleceń *ang. command line interpreter (CLI)*. Celem jej jest interpretacja wpisywanych przez użytkownika poleceń i przekazanie obsługi ich do odpowiednich aplikacji. Polecenia są w praktyce programami. Po ich zamknięciu powłoka wypisuje znak zachęty.



- ▶ Wykonaj dowolne polecenie np. `ls` wpisując je zwyczajnie w oknie terminala. *(znaczenie polecenia zostanie wyjaśnione później)*

Znak zachęty

`<user_name>@<machine>:<current_dir><prompt>`

`<user_name>` — nazwa zalogowanego użytkownika

`<machine>` — nazwa zalogowanego użytkownika

`<current_dir>` — nazwa zalogowanego użytkownika

`<prompt>` — znak zachęty np. "\$" - dla zwykłego użytkownika "#" - dla administratora

Przykład:

```
astroinski@cisco.cs.put.poznan.pl:~/pp/$ ls
```



Skróty terminala

Crtl-Alt-F1 — uruchomienie terminala tekstowego **Alt-F1 ... Alt-F2 ...**

— przełączanie się na kolejne terminale tekstowe

Crtl-Alt-F7 — powrót do środowiska graficznego

Shift-Crtl-n — otwórz nowy terminal (w środowisku graficznym)

Shift-Crtl-q — zamknij terminal (w środowisku graficznym)

Shift-LEFT — poprzedni terminal (środowisko graficzne)

Shift-RIGHT — następny terminal (środowisko graficzne)

Shift+PgUp — przewiń terminal w górę (tryb tekstowy)

Shift+PgDown — przewiń terminal w dół (tryb tekstowy)

- ▶ Przejdź do konsoli tekstowej i zaloguj się. Pozmieniaj terminale.
- ▶ Wróć z powrotem do trybu graficznego. Pobaw się podanymi skrótami.



Anatomia wywołania polecenia

```
[inf70000@unixlab ~]$ yaourt -Sy -clean docutils
```

- ▶ string zachęty
- ▶ nazwa komendy
- ▶ krótkie opcje: [-O [VALUE]]...
- ▶ długie opcje: [-OPTARG[=VALUE]]...
- ▶ argumenty: [ARG]...
- ▶ Uwagi:
 - ▶ wrażliwe na wielkość liter
 - ▶ bardzo potężne i niebezpieczne narzędzie — należy być ostrożnym korzystając z terminala
- ▶ Przykład:
 - ▶ \$ ls
 - ▶ \$ ls -l /home
 - ▶ \$ ls -a
 - ▶ \$ ls -a /home

passwd

Program pozwalający użytkownikowi na zmianę hasła. Na początku pyta o stare hasło użytkownika, a następnie prosi o dwukrotne podanie nowego. W przypadku uruchomienia bez parametru zmiana hasła dotyczy aktualnie zalogowanego użytkownika.

Składnia: `passwd [nazwa_użytkownika]`

Przykład: `$ passwd`

- ▶ zmień hasło poleceniem `passwd`
- ▶ zmień hasło poleceniem `passwd` z podaniem nazwy użytkownika



exit

Program pozwalający na opuszczenie programu, terminala oraz wylogowanie się z sieci Unix.

Składnia: `exit`

Przykład: `$ exit`

logout

Wyloguj z powłoki.

Składnia: `logout`

Przykład: `$ logout`

- ▶ Wyloguj się z systemu za pomocą jednego z wyżej wymienionych poleceń.



who

Polecenie pokazuje kto jest zalogowany do systemu.

Składnia: `who [OPTION]... [FILE | ARG1 ARG2]`

-s — wypisz tylko najważniejsze dane: nazwa, terminal, data.

-b — czas ostatniego uruchomienia systemu

...

Przykład: `$ who`

id

Polecenie pokazuje faktyczne numery UID, GID oraz grupy do których należy dany użytkownik.

Składnia: `id [options] [username]`

-g — wypisz tylko GID

-u — wypisz tylko UID

-n — razem z **-u** i **-g** wypisuje nazwy nie numer

...

Przykład: `$ id`

whoami

Wypisuje UID w postaci nazwy nie numeru. Polecenie równoważne `id -un`.

Składnia: `whoami`

Przykład: `$ whoami`



help

Dla większości programów z linii poleceń można użyć tzw. pomocy kontekstowej. Przedstawia ona podstawowe informacje jak używać polecenia. Wystarczy zaraz za nazwą polecenia dodać przełącznik `--help`.

Składnia: `<polecenie> --help`

Przykład: `$ ls --help`

- ▶ Wyświetli informacje o poleceniu `passwd`.



man

Zdecydowanie bardziej rozbudowaną pomocą jest tzw. podstawowa pomoc systemowa czyli aplikacja `man`. Funkcjonuje niemalże w każdym systemie z rodziny Unix i pomimo swojego wieku stale jest podstawowym źródłem informacji dla użytkownika systemu. Pomoc do polecenia `man` można oczywiście pokazać korzystając z polecenia `man man`. Źródłem danych dla polecenia `man` jest katalog: `/usr/share/man`.

Składnia: `man <polecenie>`

Przykład: `$ man ls`

- ▶ Wyświetl informacje o poleceniu `passwd`.

Obsługa (skrótów klawiszowe) polecenia `man`

q — wyjście

Enter, j, k — przewijanie liniami

Spacja, Crtl-f, Crtl-b, Crtl-d, Crtl-u — przewijanie stronami

g, G — przejście na początek/koniec pliku

/ — rozpoczęcie wyszukiwania tekstu

n, N — przejście do następnego/poprzedniego wystąpienia słowa

- ▶ Wyświetl informacje o poleceniu `passwd`. Następnie przewijaj stronę pomocy liniami, stronami itd. Wyszukaj słowo `pass`. Czy znajduje się wystąpienie dokładnie tego słowa?

Zmiana języka — pomocy i komunikatów

W celu uzyskania dostępu do stron pomocy w języku polskim należy ustawić zmienną środowiskową LANG w następujący sposób:

```
export LANG=pl_PL.UTF-8
```

gdzie pl oznacza język polski, PL oznacza Polskę, UTF-8 oznacza standard kodowania znaków. W przypadku problemu z wyświetlaniem niektórych znaków narodowych ustaw zmienną LANG następująco:

```
export LANG=pl_PL
```

następnie wybierz z okna terminala kodowanie ISO 8859-2 (*Settings/Encoding*).

- ▶ Przećwicz zmianę języka czytając pomoc w różnych językach, oraz wyświetlając komunikaty np. polecenia `$ rm abcdef`. Przykładowe inne słowniki to: `de_DE.UTF-8`, `fr_FR.UTF-8`, `ru_RU.UTF-8`, `zh_TW.UTF-8`.

Sekcje pomocy

Pomoc systemowa została podzielona na różne sekcje tematyczne. W każdej z sekcji mogą występować te same polecenia (w innych kontekstach), należy określić z której sekcji tematycznej chcemy korzystać. Służy do tego polecenie:

Składnia: `man <numer_sekcji> <polecenie>`

Przykład: `man 1 passwd`

Sekcje pomocy:

- ▶ **1:** polecenia
- ▶ **2:** funkcje systemowe
- ▶ **3:** funkcje biblioteczne
- ▶ **4:** pliki specjalne
- ▶ **5:** formaty plików
- ▶ **6:** gry
- ▶ **7:** różne
- ▶ **8:** polecenia administracyjne

- ▶ Odwołaj się do różnych sekcji pomocy systemowej, np.
 - ▶ `$ man sleep`
 - ▶ `$ man 3 sleep`

Uwaga!

Zapis `sleep(3)` oznacza stronę pomocy systemowej dla hasła `sleep` znajdującą się w sekcji 3.

`whatis`

Przeszukuje strony we wszystkich sekcjach tematycznych pomocy. Zwraca jednozdaniowy opis dotyczący stron w pomocy (wraz z wskazaniem sekcji) na temat wyspecyfikowanego w argumencie polecenia.

Składnia: `whatis <polecenie>`

Przykład: `$ whatis sleep`



apropos

Przeszukuje strony we wszystkich sekcjach tematycznych pomocy. Zwraca jednozdaniowy opis dotyczący stron w pomocy (wraz z wskazaniem sekcji) na temat wyspecyfikowanego w argumencie słowa. Działa bardzo podobnie do polecenia `whatis` ale przeszukuje bazę danych dopasowując słowa, a nie polecenia. Można korzystać z wyrażeń regularnych.

Składnia: `apropos <słowo>`

Przykład: `$ apropos directory`

Przykład II: `$ apropos "make.*director"`

Uwaga!

Zapis `".*"` jest fragmentem wyrażenia regularnego, oznaczającego dowolny ciąg znaków (również pusty).



- ▶ Dowiedz się gdzie można znaleźć informację w pomocy na temat polecenia `ls`?
- ▶ Dowiedz się gdzie występuje w pomocy informacja na temat metody `sleep`
- ▶ Następnie otwórz sekcje opisujące metodę `sleep`.
- ▶ Co się stanie gdy odwołasz się do strony w pomocy, której nie ma?

info

Alternatywnym programem do polecenia `man` udzielającym użytkownikowi niezbędnych informacji jest polecenie `info`. Niektóre nowe programy w stronach podręcznikowych `man` wyświetlają tylko informację, że należy otworzyć stronę pomocy za pomocą polecenia `info`.

Źródłem informacji dla polecenia jest katalog: `/usr/share/info`

Składnia: `info <polecenie>`

Przykład: `$ info sleep`

pinfo

Alternatywna przeglądarka do plików pomocy to `pinfo`.

Składnia: `pinfo <polecenie>`

Przykład: `$ pinfo sleep`



- ▶ Wykorzystując zdobytą wiedzę, na temat pomocy systemowej odszukaj informacje o następujących katalogach: `/etc`, `/bin`, `/usr`, `/tmp`, `/var`, `/home`, `/dev`. Podpowiedź: `hier(7)`.
- ▶ Korzystając z pomocy systemowej do czego służy polecenie `dir` i jakie ma możliwości.
- ▶ Przygotuj przykładową stronę pomocy systemowej do wydruku i obejrzyj ją na ekranie:
 - ▶ `$ man -t ls > out.ps`
 - ▶ `$ gv out.ps` lub `$ kghostview out.ps`
- ▶ Sprawdź do czego służą przełączniki `-p`, `-l`, `-m` polecenia `finger`.

history

Polecenie wyświetla na standardowym wyjściu historię ostatnio wykonanych poleceń.

UP — poprzednie polecenie z historii

DOWN — kolejne polecenie w historii

Crtl-r — interaktywne poszukiwanie polecenie w historii

Polecenie: \$!124 — wykonanie 124 polecenie z historii

- ▶ Wykonaj 5 polecenie od końca.
- ▶ Wykonaj 74 polecenie z historii.
- ▶ Przećwicz interaktywne wyszukiwanie w historii.

Skróty terminala

Shift-UP, Shift-DOWN — przewijanie zawartości okna terminala (tryb graficzny jak i tekstowy)

Ctrl-a, Ctrl-e — przejście na początek/koniec linii

Ctrl-f, Ctrl-b — zmiana pozycji kursora o jeden znak

Ctrl-k — usunięcie tekstu do końca wiersza

Ctrl-w — usunięcie poprzedniego słowa

Ctrl-c — przerwanie działania polecenia

Ctrl-z — uśpienie działającego procesu

Ctrl-l — oczyszczenie ekranu

Polecenie: \$ `clear` — oczyszczenie ekranu

- ▶ Przećwicz mechanizm automatycznego uzupełniania nazw programów i plików:
`$ mk<Tab><Tab>d<Tab>`
- ▶ Znajdź ten sposób wszystkie programy zaczynające się na `pr` i `x`. Uzupełnianie nazw plików przećwicz w swoim katalogu domowym:
`$ ls p<Tab>`
- ▶ Spróbuj wyświetlić zawartość katalogu `/usr/share/doc/packages`, na każdym etapie w maksymalnym stopniu wykorzystując automatyczne uzupełnianie nazw katalogów.
- ▶ Uruchom program `$ sleep 10` i przerwij jego działanie.
- ▶ Przećwicz przedstawione na poprzednim slajdzie skróty.

Nazwy uogólnione

Jest to mechanizm pozwalający na odwoływanie się z linii poleceń do bardzo uogólnionej nazwy parametru polecenia. Ma to na celu np. adresować całą grupę plików spełniających dane kryterium. * — dowolny ciąg znaków

? — pojedynczy znak

[] — pojedynczy znak ze wskazanego zbioru

Uwaga!: nazwy uogólnione to inny mechanizm niż wyrażenia regularne!

- ▶ Zastosuj nazwy uogólnione w odwołaniach do plików, stosując znaki specjalne. W celu wykonania ćwiczenia utwórz komendą `touch` puste pliki o różnych nazwach:

```
$ touch a.txt b.txt c.txt
```

```
$ touch a.dat b.dat ab.dat
```

```
$ ls *.txt
```

```
$ ls a*
```

```
$ ls *.*
```

```
$ ls [ab].txt
```

```
$ ls ?.dat
```

```
$ ls *.*???
```

- ▶ Wyświetl listę jednoznakowych programów z katalogu `/usr/bin`. Następnie wyświetl analogiczną listę trzyznakowych programów z tego katalogu. Zastosuj przełącznik `-d` komendy `ls` (zobacz dokumentacja `man`). Wyświetl programy dwuliterowe złożone z liter "a", "b" i "c". Wyświetl programy zawierające przynajmniej jedną wielką literę i jedną cyfrę.



Dzięki za uwagę!