

# **System operacyjny Linux**

Paweł Rajba

[pawel.rajba@continent.pl](mailto:pawel.rajba@continent.pl)

<http://kursy24.eu/>

# Zawartość modułu 5

---

- Wiesz poleceń
  - kontrola zadań
  - aliasy, zmienne
  - potoki, przekierowania
  - nawigacja
- Pozostałe przydatne polecenia konsolowe
- Zarządzanie procesami

# Wiersz poleceń

---

- Co to jest?
- Uruchomienie
  - ~/.bashrc
  - /etc/profile (logowania)
  - ~/.bash\_profile (logowania)
  - ~/.bash\_login (logowania)
  - ~/.profile (logowania)
- Zakończenie
  - ~/.bash\_logout

# Kontrola zadań

---

- Pojęcie terminala kontrolnego
- Uruchomienie w tle (&)
- Zatrzymanie zadania (CTRL+Z)

# Kontrola zadań

---

- jobs – listing aktualnych zadań
  - Składnia
    - jobs [-l] [-p] [numer]
  - Opcje
    - -l – wyświetla dodatkowo numery procesów,
    - -p – listuje same numery procesów,
    - numer – oznacza numer zadania — możemy wyświetlić tylko określone zadania,

# Kontrola zadań

---

- fg – przywrócenie zadania
  - Składnia
    - fg [numer]
  - Opcje
    - numer – określa numer zadania
- bg – uruchomienie w tle
  - Składnia
    - bg [numer]
  - Opcje
    - numer – określa numer zadania

# Aliasy

---

- **alias** – utworzenie aliasu
  - Składnia
    - alias [-p]
    - alias [nazwa[=wartość]]
  - Opcje
    - -p – wydrukowanie wszystkich aliasów
- **unalias** – usunięcie aliasu
  - Składnia
    - unalias [-a | nazwa]
  - Opcje
    - -a – usunięcie wszystkich aliasów

# Zmienne

---

- Wyświetlanie zmiennych
  - printenv [nazwa, ...]
- Pobranie wartości: operator \$
- Cudzysłowie i apostrofy
  - '...' – echo '\$x'
  - "..." – echo "\$x"
  - `...` – kill 'cat /var/run/named/named.pid'



# Zmienne

---

- Nadawanie wartości
  - `zmienna=wartość`
  - `cztery=2+2`
  - `cztery=${2+2}`
  - `cztery=`expr 2 + 2``
- Eksportowanie zmiennej
  - przykład
- Usuwanie zmiennej
  - `unset zmienna`

# Zmienne

---

- **Ważniejsze zmienne**
  - \$HOME — katalog domowy bieżącego użytkownika,
  - \$PATH — ścieżka poszukiwań,
  - \$UID — identyfikator bieżącego użytkownika,
  - \$PS1 — znak zachęty pierwszego poziomu,
  - \$SECONDS — liczba sekund działania powłoki,
  - \$HOSTNAME — nazwa hosta,
  - \$RANDOM — liczba losowa,
  - \$PWD — ścieżka do bieżącego katalogu.

# Potoki

---

- Zasada działania
- Przykłady
  - `finger | grep -w pawel`
  - `printenv | wc -l`
  - `ls -l /usr/bin/t* | cut -c 24-32,50-`
  - `ls -la | grep "^-" | sort -k 5 -n`

# Przekierowania

---

- >
  - `ls /bin > lista_plikow`
- >>
  - `ls /usr/bin >> lista_plikow`
- <
  - `mail -s Liscik pawel < plik_z_treścią.txt`

# Nawigacja

---

- **Kombinacje klawiszy**
  - <ctrl>a – na początek wiersza,
  - <ctrl>e – na koniec wiersza,
  - <ctrl>f – znak do przodu,
  - <ctrl>b – znak do tyłu,
  - <alt>f – słowo do przodu,
  - <alt>b – słowo do tyłu,
  - <ctrl>l – czyszczenie ekranu,
  - <ctrl>r – szukanie w historii od końca,
  - <ctrl>s – szukanie w historii od początku,
  - <ctrl>w – usuwa słowo.

# Wiersz poleceń

---

- Uzupełnianie nazw poleceń i nazw plików
  - Przykład

# Pozostałe przydatne polecenia

---

- Omówimy
  - echo, grep, cal, date, seq,
  - exit, logout,
  - help, uptime, uname,
  - cmp, cut, file, fold, head, iconv,
  - join, sort, split, tail, uniq,
  - mesg, write, wall, talk, telnet.

# Polecenia konsolowe

---

- echo – wypisuje tekst na ekran
  - Składnia:
    - echo [-neE] tekst
  - Opcje:
    - -n – pomija ostatni znak końca linii,
    - -e – włącza interpretację znaków specjalnych poprzedzonych znakiem "\",
    - -E – wyłącza interpretację znaków specjalnych poprzedzonych znakiem "\" (domyślnie).



# Polecenia konsolowe

---

- echo c.d.
  - Znaki specjalne
    - \a — alarm (bell),
    - \c — pomija ostatni znak nowej linii,
    - \n — nowy wiersz,
    - \r — powrót karetki,
    - \t — poziomy tabulator,
    - \v — pionowy tabulator,
    - \n — znak backslash-a.

# Polecenia konsolowe

---

- grep – wyszukiwanie wzorca
  - Składnia
    - grep [opcje] wzorzec [pliki]
  - Opcje
    - -e wzorzec – jawnie określa wzorzec
      - przydatne przy wzorcach zaczynających się od znaku "-",
    - -c – zlicza ilość dopasowań zamiast je wypisywać,
    - -i – włącza nierozróżnianie małych i wielkich liter,

# Polecenia konsolowe

---

- grep c.d.
  - Opcje c.d.
    - -m liczba – zatrzymuje czytanie po tylu dopasowaniach, ile określa parametr liczba,
    - -n – wypisuje dodatkowo numer wiersza, który się dopasował,
    - -r – przeszukuje pliki w podkatalogach,
    - -w – dopasują się tylko całe słowa

# Polecenia konsolowe

---

- grep c.d.
  - Wyrażenia regularne
    - c – pojedyncza litera,
    - \znak – znaki specjalne np. \\$,
    - [aAbBcC-F] – listy znaków,
    - [^abcd] – znaki spoza listy,
    - . – dowolny znak
    - ^ – początek napisu
    - \$ – koniec napisu
    - \< – początek słowa
    - \> – koniec słowa

# Polecenia konsolowe

---

- grep c.d.
  - Wyrażenia regularne
    - ? – element występuje 0 lub 1 raz,
    - \* – element występuje 0 lub więcej razy,
    - + – element występuje 1 lub więcej razy,
    - {n} – element występuje dokładnie n razy,
    - {n,} – element występuje co najmniej n razy,
    - {n,m} – element występuje co najmniej n i co najwyżej m razy,
    - r|s — oznacza dopasowanie do r lub do s,
    - rs — oznacza dopasowanie do r, a następnie do s.

# Polecenia konsolowe

---

- grep, przykład
  - wyrażenia\_regularne.txt

# Polecenia konsolowe

---

- `cal` – wyświetla kalendarz
  - Składnia
    - `cal [-3] [[month] year]`
  - Opcje
    - `-3` – wyświetla dodatkowo poprzedni i następny miesiąc

# Polecenia konsolowe

---

- `date` – wyświetla lub ustawia czas/datę
  - Składnia
    - `date [+FORMAT]` (wyświetlenie)
    - `date MMDDhhmm[YYYY][.ss]` (ustawienie)
  - Opcje formatu
    - `%a` — skrótowa nazwa dnia tygodnia (Sun..Sat),
    - `%A` — pełna nazwa dnia tygodnia (Sunday..Saturday),
    - `%b` — skrótowa nazwa miesiąca (Jan..Dec),
    - `%B` — pełna nazwa miesiąca (January..December)
    - `%d` — dzień miesiąca (01..31),



# Polecenia konsolowe

---

- date c.d.
  - Opcje
    - %e – dzień miesiąca w formacie bez wiodącego zera (1..31),
    - %H – godzina (00..23),
    - %I – godzina w formacie dwunastogodzinnym (01..12),
    - %j – dzień roku (001..366),
    - %k – godzina w formacie bez wiodącego zera (0..23),
    - %l – godzina w formacie dwunastogodzinnym i bez wiodącego zera (1..12),

# Polecenia konsolowe

---

- date c.d.
  - Opcje
    - %m – miesiąc (01..12),
    - %M – minuta (00..59),
    - %s – liczba sekund od "00:00:00 1970-01-01 UTC"
    - %S – sekunda (00..59),
    - %u – dzień tygodnia (1..7); 1 oznacza poniedziałek (Monday),
    - %Y – rok (od 1970).

# Polecenia konsolowe

---

- seq – generowanie sekwencji liczb
  - Składnia
    - seq [opcje] [pierwsza [krok]] ostatnia
  - Opcje
    - -s napis – napis będzie oddzielał kolejne liczby
    - -w – liczby będą tej samej szerokości (zostaną dodane wiodące zera)

# Polecenia konsolowe

---

- help – pomoc dotycząca poleceń powłoki
  - Składnia
    - help [-s] [wzorzec]
  - Opcje
    - -s – wyświetli skrócony opis
- uptime – informuje o czasie działania systemu
  - Składnia
    - uptime

# Polecenia konsolowe

---

- **uname** - wyświetla informacje o systemie
  - Składnia
    - `uname [opcje]`
  - Opcje
    - `-a` – wyświetla wszystkie informacje.
    - `-s` – wyświetla wersje systemu operacyjnego,
    - `-n` – wyświetla nazwę hosta,
    - `-r` – wyświetla wersje jądra,
    - `-m` – wyświetla informacje o sprzęcie,
    - `-o` – wyświetla nazwę systemu operacyjnego.

# Polecenia konsolowe

---

- `cmp` – porównuje dwa pliki tekstowe
  - Składnia
    - `cmp [opcje] plik1 plik2 [skip1 [skip2]]`
  - Opcje
    - `-b` – wypisuje pierwszy znak różnicy
    - `-n limit` – porównuje tylko limit znaków
    - `skip1, skip2` – po ilu znakach będzie rozpoczynał porównanie pierwszego, drugiego pliku

# Polecenia konsolowe

---

- cut – pobiera żądany fragment z wiersza
  - Składnia
    - cut [opcje] [pliki]
  - Opcje
    - -c lista – drukuje znaki określone w liście lista,
    - -f lista – drukuje pole określone w liście lista,
    - -d lista – określa znak rozdzielający pola i domyślnie jest to znak tabulatora; używamy go w połączeniu z opcją -f.

# Polecenia konsolowe

---

- cut c.d.
  - Zakresy na liście
    - N – N-ty bajt, znak lub pole,
    - N- – bajty, znaki lub pola od N-tego do ostatniego,
    - N-M – bajty, znaki lub pola od N-tego do M-tego,
    - -M – bajty, znaki lub pola od pierwszego do M-tego.



# Polecenia konsolowe

---

- file – rozpoznaje rodzaj zawartości pliku
  - Składnia
    - file [-zp] lista-plików
    - file [-zp] -f nazwa-pliku
  - Opcje
    - -z – będzie zaglądał do spakowanej zawartości
    - -p – będzie zachowywany dotychczasowy czas dostępu
    - -f plik – podajemy plik z listą plików do weryfikacji
  - Przykład
    - tworzymy plik tekstowy, plik tar.gz i sprawdzamy poleceniem file

# Polecenia konsolowe

---

- fold – zawija wiersze z pliku
  - Składnia
    - fold [-s] [-w szerokosc] plik
  - Opcje
    - -s – będzie łamał na spacji, a nie w środku słowa
    - -w szerokosc – ustalamy inną szerokość wiersza niż 80

# Polecenia konsolowe

---

- head – pobiera początkowy fragment pliku
  - Składnia
    - head [-c [-]N] [-n[-]N] [-N] plik
  - Opcje
    - -c N – pobiera pierwszych N znaków
    - -c -N – pobiera wszystkie znaki oprócz N ostatnich
    - -l N – pobiera pierwszych N wierszy
    - -l -N – pobiera wszystkie wiersze oprócz N ostatnich

# Polecenia konsolowe

---

- `iconv` – konwertuje kodowanie plików
  - Składnia
    - `iconv --list`
    - `iconv -f kodowanie -t kodowanie plik [-o plik]`
  - Opcje
    - `-f kodowanie` – kodowanie pliku źródłowego
    - `-t kodowanie` – kodowanie pliku docelowego
    - `--list` – drukuje listę dostępnych stron kodowych
    - `-o plik` – przekierowuje wynik do pliku (zamiast drukować na ekran)
  - Przykład
    - `iconv --help | iconv -f UTF-8 -t ISO_8859-2`

# Polecenia konsolowe

---

- join – łączy pliki wg wspólnej kolumny
  - Składnia
    - join [-1 pole] [-2 pole] [-j pole] [-i] [-o format] plik1 plik2
  - Opcje
    - -1 pole1, -2 pole1 – do łączenia z plik1 będzie używana kolumna pole1, a z plik2 – pole2; domyślnie pierwsze k.
    - -j pole – skrót na -1 pole -2 pole
    - -i – przy porównywaniu nie będą brane pod uwagę wielkości liter
    - -o format – określa format wyniku; jest to lista o elementach postaci: NUMER.POLE

# Polecenia konsolowe

---

- **sort** – sortuje zawartość plików
  - Składnia
    - `sort [-n] [-r] [-k k1, k2, ...] [pliki]`
  - Opcje
    - `-n` – dane wartości będą porównywane numerycznie, czyli będą traktowane jako liczby,
    - `-r` – odwrotny porządek sortowania,
    - `-k k1, k2, ...` – kolumny, wg których ma być wykonane kolejno sortowanie; jeśli nie podamy tego parametru, sortowanie będzie wykonane wg pierwszej kolumny.
    - `-u` – zostaną usunięte powtarzające się wiersze

# Polecenia konsolowe

---

- split – dzieli plik na mniejsze
  - Składnia
    - split [opcje] plik [prefiks]
  - Opcje
    - -a N – sufixs będzie długości N (domyślnie 2)
    - -b N – do plików będzie ładowane po N bajtów
    - -d – sufixsy będą liczbowe, a nie literowe
    - -l N – do plików będzie ładowane po N wierszy
  - Komentarz
    - domyślnie: prefiksem jest 'x' i do plików ładowane jest po 1000 wierszy

# Polecenia konsolowe

---

- tail – pobiera końcowy fragment pliku
  - Składnia
    - tail [-c N] [-n N] [-f] plik
  - Opcje
    - -c N – pobiera ostatnich N znaków
    - -l N – pobiera ostatnich N wierszy
    - -f – drukuje na bieżąco przyrost pliku
    - -s – łączymy z -f, określa odstępy czasu pomiędzy odświeżaniem zawartości pliku na ekranie



# Polecenia konsolowe

---

- `uniq` – usuwa powtarzające się wiersze z pliku
  - Składnia
    - `uniq [-c] [-d | -u] [-i] plik`
  - Opcje
    - `-c` – obok wiersza zostanie wydrukowana ilość powtórzeń
    - `-d` – drukowane będą wiersze powtórzone co najmniej raz
    - `-u` – drukowane będą wiersze, które się nie powtarzają
    - `-i` – przy porównywaniu wielkość znaków nie będzie miała znaczenia

# Polecenia konsolowe

---

- mesg – włącza/wyłącza możliwość otrzymywania komunikatów
  - Składnia
    - mesg [n] [y]
  - Opcje
    - n – wyłącza
    - y – włącza
  - Komentarz
    - ustawia prawo dostępu do pisania na konsolę  
np. plik /dev/pts/1

# Polecenia konsolowe

---

- write – wysyła komunikat do użytkownika
  - Składnia
    - write user [terminal]
  - Uwagi
    - po zakończeniu wpisywania tekstu wprowadzamy kombinację <ctrl>-d
- wall – wysyła komunikat do wszystkich użytkowników
  - Składnia
    - wall

# Polecenia konsolowe

---

- talk – służy do rozmowy interaktywnej
  - Składnia
    - talk [-p kodowanie] uzytkownik [terminal]
  - Opcje
    - -p kodowanie - określamy kodowanie tekstu wysyłanego przez rozmówcę,
    - uzytkownik – nazwa użytkownika; jeśli jest na zdalnym komputerze, używamy formatu uzytkownik@host
    - terminal – określamy terminal, na którym chcemy z kimś rozmawiać

# Polecenia konsolowe

---

- telnet – służy do zdalnego łączenia się
  - Składnia
    - telnet [-l użytkownik] [host [port]]
  - Opcje
    - -l użytkownik – logowanie jako określony użytkownik,
    - host – określa zdalny komputer,
    - port – określa zdalny port połączenia.

# Zarządzanie procesami

---

- Wprowadzenie do procesów
  - proces to uruchomiony program
  - zarządzania procesem jest związane przydziałem pamięci, czasu procesora i pamięci dyskowej
  - jeden procesor obsługuje tylko jeden proces
  - przełączanie jako mechanizm umożliwiający stworzenie wrażenia równoległości

# Zarządzanie procesami

---

- Wybrane parametry procesów
  - PID – identyfikator procesu
  - PPID – identyfikator procesu macierzystego
  - UID – identyfikator użytkownika, który utworzył proces
  - EUID – określa, do których zasobów proces ma prawa dostępu; zwykle UID=EUID

# Zarządzanie procesami

---

- Wybrane parametry procesów c.d.
  - GID, EGID – parametry analogiczne do UID i EUID, tylko dotyczące grup
  - Priorytet – określa priorytet procesu; wartości od -20 (najwyższy) do 19 (najniższy)
  - Terminal – określa terminal kontrolny, czyli m.in. określa dla procesu stdin, stdout, stderr.



# Zarządzanie procesami

---

- Cykl życia procesu
  - proces jest zawsze tworzony przez inny proces jako efekt podziału (funkcja fork)
  - zwykle po podziale nowy proces uruchamia nowy program (funkcja exec)
  - proces dostaje swoją pamięć, czas procesora, itp. i zaczyna działać

# Zarządzanie procesami

---

- Cykl życia procesu c.d.
  - kiedy proces chce skończyć swoje działanie, informuje o tym jądro
    - zwalniane jest zasoby związane z tym procesem
    - proces jest w stanie zombie
  - za ostateczne zakończenie procesu odpowiada proces macierzysty

# Zarządzanie procesami

---

- Sygnały
  - są sposobem na komunikację z procesem
  - proces po otrzymaniu sygnału wykona
    - pewną domyślną procedurę obsługi sygnału
    - zarejestrowaną dla tego sygnału procedurę jego obsługi
  - sygnałów jest ok. 30

# Zarządzanie procesami

---

- Główne stany procesów
  - wykonywany (R) – proces wykonywany
  - uśpiony
    - przerywalnie (S) – czeka na zakończenie się jakiegoś zdarzenia, np. dostęp do zasobu
    - nieprzerywalnie (D) – również czeka, ale coś innego (zwykle do dostęp urządzeń WE/WY)
  - zatrzymany (T) – proces ma zakaz wykonywania
  - zombie (Z) – proces próbuje się zakończyć; proces jest zatrzymany, ale nie przechwycony przez proces macierzysty

# Zarządzanie procesami

---

- Polecenia do obsługi procesów
  - KILL – wysyłanie sygnałów do procesów
  - NICE, RENICE – zmiana priorytetów
  - PS, TOP – monitorowanie procesów

# Zarządzanie procesami

---

- kill – wysyłanie sygnałów do procesów
  - Składnia
    - kill [-s SYGNAŁ | -SYGNAŁ] PID
    - kill -l [SYGNAŁ]
  - Opis
    - wysłanie sygnału
    - informacje o sygnałach

# Zarządzanie procesami

---

- nice – uruchamianie programu z ustawieniem priorytetu
  - Składnia
    - nice -n priorytet polecenie argumenty
  - Opcje
    - -n priorytet – liczba ustalająca priorytet, która powstaje do dodaniu do bieżącego priorytetu użytkownika,
    - zwykły użytkownik może podawać wartości tylko  $\geq 0$
  - Uwaga
    - jeśli uruchomimy nie podając opcji -n i domyślnym priorytetem użytkownika jest 0, proces dostanie priorytet 10

# Zarządzanie procesami

---

- nice, przykład
  - tworzymy program p2.cpp o następującym kodzie
    - ```
#include <stream>
using namespace std;
int main() { int i=0; int j=0; while (j<40) { if ( i<1000000000 )
i++; else { i=0; cout << "Gucio " << j++ << endl; } } }
```
  - kompilujemy g++ i uruchamiamy dwa razy
    - nice -n 0 ./a.out
    - nice -n 5 ./a.out



# Zarządzanie procesami

---

- renice – zmiana priorytetu procesu
  - Składnia
    - renice priorytet [-p PIDS] [-u USERS]
  - Opcje
    - priorytet – nowy priorytet
    - -p – lista nr. procesów np. 1400 2300 123
    - -u – lista użytkowników np. pawel zenon
  - Przykłady użycia
    - renice 0 -u pawel -p 15517
    - renice 5 -p 15517 15518

# Zarządzanie procesami

---

- renice, przykład
  - tworzymy program o następującym kodzie
    - `int main() { while (1) { int x=7; int y=x*x*x*x*x; } }`
  - kompilujemy i uruchamiamy w tle 5 razy
  - ustalamy PID jednej z uruchomionych kopii
    - raz zmieniamy nice na -20, odpalamy top i patrzymy na czas procesora
    - drugi raz zmieniamy nice na 19, odpalamy top i znowu patrzymy na czas procesora

# Zarządzanie procesami

---

- ps – drukuje informacje o procesach
  - Składnia
    - ps [opcje]
  - Opcje określające, które procesy będą wypisane
    - -A, -e – drukuje wszystkie procesy,
    - a – drukuje wszystkie procesy posiadające terminal kontrolny (nie tylko bieżącego użytkownika),
    - r – drukuje procesy aktualnie uruchomione,
    - -U user1,user2,... – drukuje procesy użytkowników user1, user2, ...
    - -C nazwa – drukuje procesy, które zostały uruchomione przez polecenie nazwa.

# Zarządzanie procesami

---

- ps c.d.
  - Opcje określające, jak listing będzie wyglądał
    - u – format użytkownika
    - l – długi format
    - w, ww – drukuje pełne (obcięte lub nie) polecenia, które uruchomiły dany proces
    - -o format – drukuje wg formatu
  - Przykłady
    - ps a -o user,ppid,pid,rss,tty,state,comm
    - ps a -o "%U %P %p %y %z %c"
    - ps -C sshd -o pid=

# Zarządzanie procesami

---

- ps c.d.
  - Niektóre kolumny wyniku z polecenia ps
    - USER – nazwa właściciela procesu,
    - PID – identyfikator procesu,
    - %CPU – % wykorzystania procesora przez proces,
    - %MEM – % wykorzystania pamięci przez proces,
    - VSZ – ilość pamięci wirtualnej przydzielonej procesowi,
    - RSS – ilość pamięci fizycznej wykorzystywanej przez proces,
    - TTY – terminal kontrolny; ? – brak terminala

# Zarządzanie procesami

---

- ps c.d.
  - Niektóre kolumny wyniku z polecenia ps
    - STAT – stan procesu, gdzie najczęściej spotykanymi stanami są
      - S (proces uśpiony)
      - R (proces aktualnie wykonywany),
    - START – godzina lub data uruchomienia procesu,
    - TIME – przedział czasu wykorzystany przez CPU,
    - COMMAND – polecenie procesu wraz z parametrami,
    - NICE – priorytet procesu,
    - PPID – identyfikator procesu rodzica.

# Zarządzanie procesami

---

- top – interaktywna lista procesów
  - Składnia
    - top [opcje]
  - Opcje
    - -d liczba – liczba określa, co ile sekund należy odświeżać listę procesów,
    - -n liczba – top wyłączy się po odświeżeniu danych liczba razy,
    - -u user – wyświetla procesy użytkownika user.

# Zarządzanie procesami

---

- top c.d.
  - Komendy po uruchomieniu
    - d – ustawia częstość odświeżania,
    - u – ustawia użytkownika, którego procesy będą wyświetlane,
    - c – wyświetla całe polecenie dla procesu,
    - x – włącza/wyłącza podświetlanie kolumny, według której jest sortowanie,
    - y – włącza/wyłącza podświetlanie procesów (wierszy), które są uruchomione,
    - i – wyświetla wszystkie/tylko uruchomione procesy,
    - z – włącza/wyłącza kolory wyświetlanie w kolorach.