

Narzędzia Wspierające Programowanie

Bash – poszerzenie podstaw

Zasiadamy przy terminalu Linuxowym. Sprawdźmy **podstawowe cechy i zasoby**.

Ilość pamięci:	\$ free -h
Miejsce na dysku:	\$ df -H .
Miejsce na koncie:	\$ quota -s -u {MyLogin}
Nazwa dystrybucji Linuxa:	\$ hostnamectl
Ilość wątków CPU:	\$ nproc

Dodatek

Adres IP komputera:	\$ hostname -I
Adres DNS komputera:	\$ host `hostname -I`
Informacje o CPU:	\$ less /proc/cpuinfo (wyjście: q)

Wypis zawartości katalogu – usprawnijmy to.

Samo `ls` wypisuje tylko nazwy, a `ls-l` ma długi wypis i słabo rozróżnia rodzaje.

→ Propozycja: \$ `ls -ogh --color`

Krótszy wypis, czytelne długości plików, koloryzacja ścieżek, ...
(*komenda jest długa – zaradzimy temu przy konfiguracji bash'a*)

Wypiszmy tylko ścieżki: \$ `ls -d */`

Wypis plików ukrytych: \$ `ls -a` (pliki o nazwach zaczynających się od `.`)

Wypis w formie drzewa: \$ `tree`

Kopiowanie i przenoszenie

Aby w kopii zachować datę: \$ cp -p (pomaga zrozumieć chronologię)

Kopiowanie ścieżek \$ cp -r

Przenoszenie ścieżek \$ mv -f

Kopiowanie z paskiem postępu \$ gcp (przy dużych danych – szacujemy postęp)

Poszukiwanie plików

Aby wyszukać plik(i): \$ find {od/sciezki} -name {plik/i}

Np.: \$ find . -name mojPlik.*

Aby wyszukać ścieżki: \$ find {od/sciezki} -type d -name {ścieżka/i}

Wyszukiwanie fraz w środku plików

Aby wyszukać frazy: \$ grep {fraz} {plik}

Często jednak szukamy frazy, nie wiedząc, w którym jest pliku, a np. jest wiele podkatalogów.

→ Propozycja: \$ grep --color -inr --include=*. {C,h} fraza

gdzie:	-r (recursive)	szukaj w podkatalogach
	-i (ignore case)	akceptuj każdą wielkość liter
	-n (line number)	wypisz nr linii, gdzie jest fraza
	--include={type}	szukaj wśród plików tego typu

(komenda jest długa – zaradzimy temu przy konfiguracji bash'a)

Linki symboliczne

Jeśli plik ma występować w kilku katalogach lub pod kilkoma nazwami, to skopiowanie go zabierze miejsce na dysku. Lepiej utworzyć link symboliczny do oryginału.

Link do pliku, adres względny: `$ ln -s {ścieżka_stąd}/plik_oryginalny .`

Gdy plik jest na ścieżce obok naszej: `$ ln -s ../{ścieżka_obok}/plik_oryginalny .`

Podanie pełnej lokalizacji: `$ ln -s /pełna/ścieżka/do/plik_oryginalny .`

Wypis linku w katalogu: `$ ls -l {link_symboliczny}`

Wypis tylko linków w katalogu: `$ ls -l | grep ^l`

Dodatek

Jak szybciej pisać w terminalu dzięki **skrótom klawiszowym**:

[Ctrl Shift C/V]	copy/paste w terminalu
[Ctrl ←/→]	w lewo/prawo o słowo
[Ctrl A]	skok na początek linii
[Ctrl E]	skok na koniec linii
[Ctrl U]	kasuj od kursora w lewo
[Ctrl K]	kasuj od kursora w prawo
[Ctrl D]	kasuj znak pod kursorem
[Ctrl W]	kasuj od kursora do początku słowa
[Alt D]	kasuj od kursora do końca słowa
[Ctrl T]	zamień literę pod kursorem z poprzednią
[Ctrl R]	podpowiedzi z poprzednich komend

Szersze zestawienie skrótów klawiszowych i pomocniczych komend np. tu: [Link](#)

Czytniki plików tekstowych

Często chcemy na szybko przejrzeć plik (np. z danymi – wtedy plik może być b. długi).
Wciąganie długiego pliku do edytora długo trwa. ⇒ Warto znać szybkie czytniki tekstowe.
Również, czytnik zabezpiecza przed nieumyślną zmianą treści.

- `less {plik}` : Szybki (nawet b. długi plik otworzy się w moment)
 Brak koloryzacji składni kodów

`less -N {plik}` numeruj linie

<code>PgDn</code> , <code>PgUp</code>	skok do następnej / poprzedniej strony
<code>g</code> , <code>G</code>	skok na koniec / początek pliku
<code>/fraza</code>	szukaj wystąpienia frazy
<code>n</code> , <code>N</code>	szukaj następne/poprzednie wystąpienie
<code>q</code>	wyjście

- `view {plik}` : Czytnik będący obcięciem edytora `vim` do trybu `read-only`
 Szybki (długie pliki otwiera w moment)
 Koloryzacja składni kodów (ułatwia lekturę)

<code>PgDn</code> , <code>PgUp</code>	skok do następnej / poprzedniej strony
<code>[Shift g]</code> , <code>gg</code>	skok na koniec / początek pliku
<code>/fraza</code>	szukaj wystąpienia frazy
<code>n</code> , <code>N</code>	szukaj następne/poprzednie wystąpienie
<code>:set nu</code>	numeruj linie
<code>:q</code>	wyjście

Dodatek

- `nano {plik}` : Edytor prosty w obsłudze, Koloryzacja składni kodów.
 Możliwość edycji kilku plików na raz

<code>[Ctrl+W]</code> <code>fraza</code>	szukaj frazy
<code>[Ctrl+W]</code>	szukaj kolejnego wystąpienia
<code>[Alt+B]</code>	tryb szukania wstecz
<code>[Ctrl W]</code> <code>[Ctrl T]</code> <code>{nr}</code>	skocz do linii nr.
<code>[Ctrl K]</code>	skasuj aktualną linię, ale zapamiętaj ją w schowku
<code>[Ctrl U]</code>	wstaw tu linię ze schowka
<code>[Alt Shift #]</code>	numeruj linie
<code>[Ctrl R]</code>	otwórz nowy plik w nowym „buforze”
<code>[Alt >]</code> <code>[Alt <]</code>	przeskok do edycji następnego / poprzedniego pliku

Typowy kompresor plików	gzip {pliki} → {pliki}.gz		
i dekompresja:	gunzip {pliki}.gz lub gzip -d {pliki}.gz		
Pakowacz plików:	tar {archiwum} {plik1} {plik2} ... {również wildcards}		
Typowe pakowanie:	tar czvf {archiwum.tgz} {pliki źródłowe}		
	gdzie:	c (create) z (zip) f (file)	utwórz archiwum skompresuj dotyczy plików
Wypis zawartości archiwum	tar tf {archiwum}		
Typowe rozpakowywanie	tar xzvf {archiwum.tgz}		
	gdzie:	x (extract)	wypakuj
Przy Big Data:			
Wielowątkowy kompresor:	pigz {pliki} lub pigz -n {L. wątków} {pliki}		
Połączenie tar i pigz	tar -c -I pigz -f archive.tgz {pliki_zrodlowe}		
Zdalne kopiowanie			
Ściągnięcie pliku z internetu	wget {url}		
Kopia z/na zdalny komputer	scp lub rsync		
→ Propozycja:	rsync -avzP {źródło} {login@node:ścieżka/od/home}		
	gdzie:	a (archive) z (zip) P (partial+progress)	zachowa właściwości plików kompresja w locie (szybciej) pokazuje postęp kopiowania + przy przerwaniu pozwala wznowić
rsync sprawdza, czy części paczki nie ma już w miejscu docelowym. Kopiuje, gdy ich nie ma lub gdy plik docelowy jest starszy (uzysk czasu, przydatne do backupu)			

Pliki konfiguracyjne basha:

- ~/ .bashrc wywoływany podczas logowania (ssh lub otwarcie desktopu)
- ~/ .bash_login wywoływany np. przy nowym terminalu (w ramach tego samego logowania)

Przykładowy, prosty plik ~/ .bash_login

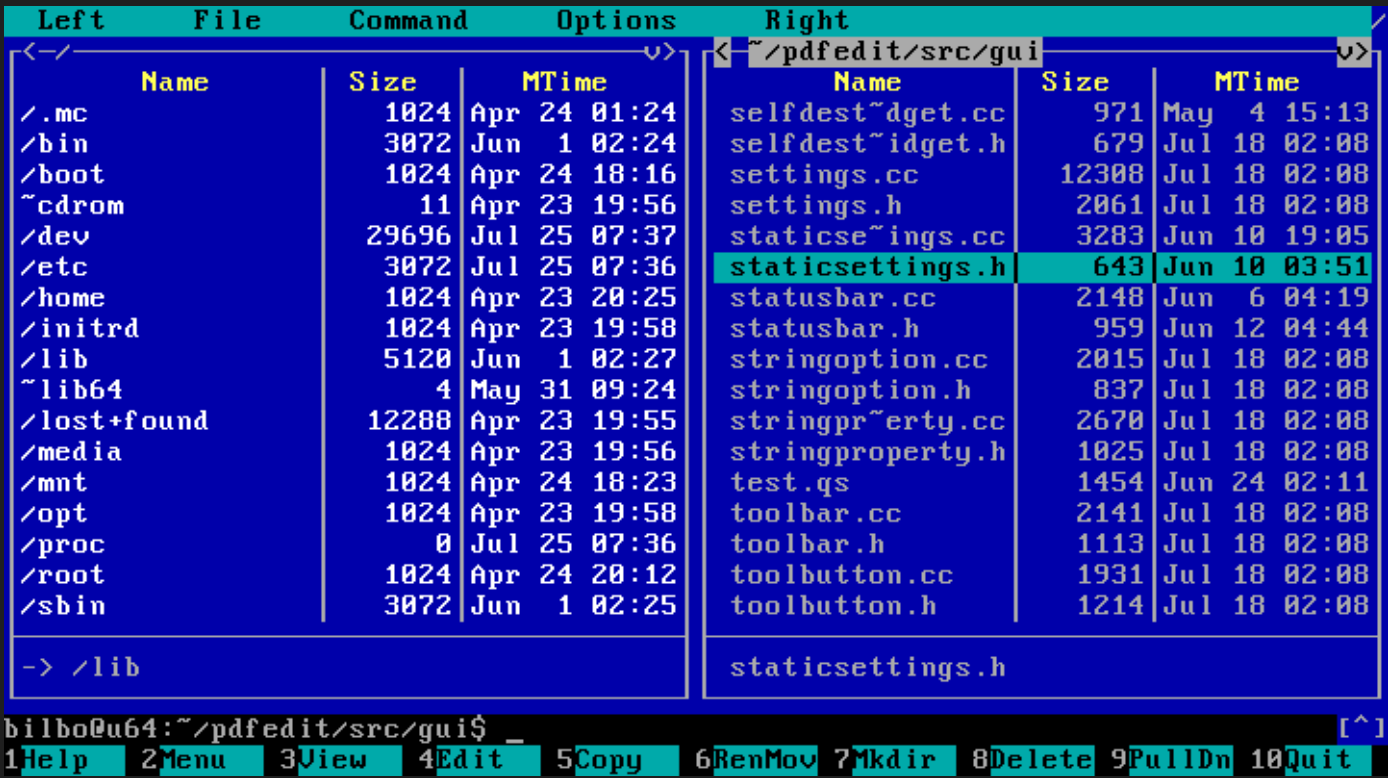
```
#!/bin/bash
echo Running ~/ .bash_login
alias ls="ls --color"
alias fn="find . -name"
alias cgrep="grep --color -inr --include=*. {C,cpp,cxx,c,h,hpp}"
alias rsync="rsync -avzP"
```

Nadajmy mu prawo do wykonania: \$ chmod 755 ~/ .bash_login

Dla pewności, że się uruchomi, do .bashrc dodajmy na koniec:
 . ~/ .bash_login

Dodatek:

- mc (Midnight Commander) - tekstowa przeglądarka plików



- [Tab] Przeskok pomiędzy oknami
- [F3] Czytnik pliku
- [F8] Skasuj plik (ratuje, gdy plik ma w nazwie trudne znaki kontrolne)
- [F10] Wyjście