Informatyka

MPDI2 sem. zimowy

wykład 30 h

lab. 30h

egzamin!

dr inż. Tomasz Bajorek

WBMiL, Zakład Informatyki, Bud.L, pok.28

e-mail: tbajorek@prz.edu.pl

12 wykładów

2 testy wykładowe

termin rezerwowy (poprawa testów)

EGZAMIN (stacjonarny) w sesji

warunki zwolnienia z egzaminu

Wykład - skrót tematyki:

System Linux – operacje na plikach i katalogach, prawa dostępu, dowiązania. Edytor *vi*.

Język znaczników HTML. Rozszerzony hipertekst: CSS

Języki programowania (składnia, semantyka). Idea programowania strukturalnego. Program i jego składowe. Struktura prostego programu. Stałe, zmienne. Proste typy danych, operacje. Zmienne łańcuchowe. Operatory logiczne, relacyjne. Instrukcje proste, instrukcje strukturalne (warunkowe, iteracyjne) - definicje, przykłady zastosowań. Generator losowy, obliczenia statystyczne.

Język Javascript - obliczenia arytmetyczne, rola zmiennych, wyprowadzanie obliczonej wartości, instrukcja warunkowa i iteracje, funkcje użytkownika.

Algorytmy i sposoby ich zapisu (pseudokod, schematy blokowe, kod), analiza poprawności i optymalizacja algorytmów, złożoność algorytmów. Algorytmy sortowania i wyszukiwania danych, algorytmy iteracyjne i rekurencyjne.

Matlab (rozszerzenie), funkcje użytkownika, operacje plikowe, obliczenia symboliczne.

Strukturalne typy danych: tablice, rekordy.

Typ obiektowy, charakterystyka, programowanie dla GUI, programy komponentowe: wykorzystanie pól i metod komponentów, programowanie zdarzeń.

Dynamiczne struktury danych: listy, tablicowe implementacje list, stos, kolejki, sterty, drzewa i ich reprezentacje, implementacje struktur dynamicznych przy pomocy tablic.

Typ zbiorowy - operacje teoriomnogościowe.

Procedury, funkcje i moduły. Rekurencja.

Literatura

William E. Shotts Jr. **:Linux. Wprowadzenie do wiersza poleceń,** Helion

Aho A. V., Hopcroft J. E., Ullman J. D.: **Algorytmy i struktury danych**, Helion, Gliwice, 2003

Freeman A.:**HTML5. Przewodnik encyklopedyczny,** Helion 2013

Lis M**., JavaScript. Ćwiczenia praktyczne**. Wydanie II, Helion,

Mrozek B., Mrozek Z., MATLAB i Simulink. Poradnik użytkownika, Helion 2010

Pratap R., **Matlab** dla naukowców i inżynierów

Bajorek T., **MATLAB - podstawy użytkowania z przykładami**, PRz

Dostęp do plików *pdf* wykładów i instrukcji do laboratorium w sieci:

adresy url:

http://tbajorek.prz.edu.pl (login: student hasło: samoloty)

http://tbajorek.v.prz.edu.pl (autoryzacja jak w USOS)

Wykład 1

System operacyjny Linux

Unix system operacyjny napisany w 1969 r. w Bell Labs

- Linux – należy do rodziny "uniksopodobnych" systemów operacyjnych opartych o jądro Linux (np. Solaris, FreeBSD i inne)

Linux jest jednym z przykładów wolnego i otwartego oprogramowania (open source)

Rola podstawowa – sieciowy system plików, usługi sieciowe (**serwery**), ale może także posiadać aplikacje użytkowe.

Są nakładki "okienkowe" ułatwiające administrację

Serwer – scentralizowany komputer świadczący usługi dla innych

- magazyn plików, m.in. dokumentów HTML,
- aplikacji
- mechanizmy udostępniania

Modele architektury komunikacyjnej

klient-serwer – – rozdzielenie funkcji komputera żądającego i komputera świadczącego usługi

P2P (od ang. peer-to-peer – równy z równym) – model komunikacji bezpośredniej komputerów – każdy może pełnić rolę klienta lub serwera

Klient/serwer – asymetryczna architektura oprogramowania w celu zwiększenia elastyczności, ułatwienia wprowadzania zmian w każdej z części. Serwer zapewnia usługi dla klientów, którzy mogą komunikować się z serwerem wysyłając żądanie (request). Np. serwer pocztowy, serwer WWW, serwer plików, serwer aplikacji. Z usług jednego serwera może zazwyczaj korzystać wielu klientów, jeden klient może korzystać jednocześnie z usług wielu serwerów.

P2P- gwarantuje obydwu stronom równorzędne prawa. Każdy komputer może jednocześnie pełnić zarówno funkcję klienta jak i serwera.

Implementacje modelu P2P: jaką są programy do wymiany plików w Internecie (Napster, eDonkey, eMule – czasem serwery katalogują pliki do wymiany), także Skype (protokół UDP), komunikatory

Uwaga: Ochrona praw autorskich przy wymianie plików

Typy architektury klient/serwer:

- architektura dwuwarstwowa – przetwarzanie i składowanie danych odbywa się w jednym module

np. przeglądarka klienta (1 warstwa), żąda strony statycznej od serwera HTTP (2 warstwa)

- architektura trójwarstwowa – przetwarzanie i składowanie danych następuje w dwóch osobnych modułach

np. przeglądarka klienta (1 warstwa), żąda od serwera HTTP (2 warstwa) , a ten współpracuje z bazą danych SQL (3 warstwa) – czyli serwer HTTP jest jednocześnie klientem serwera SQL

- architektura wielowarstwowa – przetwarzanie, składowanie i inne operacje na danych odbywają się w wielu osobnych modułach.

Zalety

- **wszystkie informacje przechowywane są na serwerze co zapewnia bezpieczeństwo danych.**
- **serwer może decydować kto ma prawo do odczytywania i zmiany danych.**

Wady

- **przepustowość (duża liczba klientów)**
- **awaria serwera – brak usług dla wszystkich klientów**

System UNIX jest:

- wielodostępny
- wielozadaniowy

czyli może obsługiwać jednocześnie wielu użytkowników i wykonywać jednocześnie wiele zadań

System serwerowy:

- wspólne pliki,
- zdalne oprogramowanie użytkowe
- serwer www
- serwer poczty e-mail
- serwer baz danych

Istnieje wiele systemów „unixopodobnych”:

- Linux
- Iris

Linux ma też wiele tzw. dystrybucji:

- Ubuntu
- Fedora
- Debian
- RedHat

i inne

System LINUX składa się z:

- **jądra,**
- **powłoki**
- **z wielu podsystemów** i programów zapewniających określone usługi np. obsługę systemu plików, pocztę, dostęp do stron www, transakcji, płatności, obsługi urządzeń itp.

Jądro (ang. kernel)

Jądro zawiera zbiór programów - zarządzanie zasobami. Jądro ma kontrolę nad komputerem, a użytkownik komunikuje się z jądrem przez tzw. **powłokę**.

Powłoka (ang. shell)- dostęp do jądra systemu, istnieje wiele powłok (języków powłok) - powłoka Bourne'a (sh), powłoka Korna (ksh), powłoka C (csh)

Po zalogowaniu (zarejestrowanego użytkownika – login/hasło), system operacyjny przenosi użytkownika do katalogu osobistego o nazwie odpowiadającej loginowi (ang. home directory – podrzędnego do katalogu o nazwie home, który grupuje wszystkich użytkowników), uruchamiany jest program powłoki w oczekiwaniu na polecenia użytkownika w postaci tekstowej.

Powłoka analizuje poprawność i możliwość realizacji poleceń i przekazuje polecenia użytkownika do jądra.

Praca zdalna na komputerze odległym

Umożliwiają ją programy komunikacyjne:

- telnet
- putty

i inne

Autoryzacja dostępu (login i hasło)

Bezpieczeństwo - protokoły szyfrowania transmisji (np. SSH)

Jak przećwiczyć pracę na serwerze linuxowym?

- laboratorium
- Niektóre serwery umożliwiają nieautoryzowany dostęp (login : **anonymous** bez hasła) – szukajcie w google
- Internetowe symulatory terminala linuxa –

np. webminal.org (rejestracja i terminal) - powłoka Bourne'a

Po zalogowaniu system zgłasza się wierszem poleceń .

```
[jan_kowalski ~]$ tu wpisujemy polecenia
```

Jeśli polecenie jest błędne otrzymamy o tym fakcie informację (ang.)

System plików w Linux-ie

Podobny do Windows – jeden katalog główny (nieusuwalny)

i drzewiasta, wspólna! struktura katalogów

Jeśli serwer ma kilka dysków fizycznych (także napędów optycznych i innych pamęci zewn.) nie są one symbolizowane jak w Windows literami - mogą być reprezentowane jako osobne katalogi (montowanie - *mount*)

/	symbol katalogu głównego (bez nazwy)
.	symbol katalogu bieżącego
..	symbol katalogu nadrzędnego (dwie kropki)
~	symbol katalogu domowego użytkownika

Ważniejsze katalogi systemowe

/bin	binarne (wykonywalne) pliki najbardziej podstawowych narzędzi systemowych
/boot	pliki niezbędne do uruchomienia systemu
/dev	za pośrednictwem tu umieszczonych plików system komunikuje się z urządzeniami
/etc	pliki konfiguracyjne, ustawienia systemowe
/home	pliki określające ustawienia każdego użytkownika,
/lib	systemowe biblioteki współdzielone (shared **lib**raries),
/mnt	tu są "montowane" dyski
/root	ustawienia użytkownika *root* - głównego administratora każdego systemu uniksowego, który ma maksymalne uprawnienia
/sbin	pliki wykonywalne poleceń, które mogą być wykonywane tylko przez administratora
/tmp	pliki tymczasowe
/usr	dodatkowe programy
/var	pliki systemowe, ale których zawartość się zmienia, jak logi programów/systemu, pliki html czy skrypty php wykorzystywane przez serwer www - dane zapisywane przez system i ważniejsze programy

Nazwy plików i katalogów

- odróżnialne małe i duże litery! czyli mogą być dwa pliki o nazwach x i X (w Windows nie), Windows i Linux nie pozwolą na plik i katalog o tej samej nazwie)
- mogą się zaczynać od cyfry, niektóre znaki różne od cyfr i liter są dozwolone, np. _, kropka (może być wiele kropek w nazwie)
- pojęcie rozszerzenia nazwy (o funkcjonalności jak w Windows) nie istnieje
- nazwa zaczynająca się od kropki to plik ukryty
- nie wolno używać SPACJI! wewnątrz nazwy

System "widzi" nazwy plików i katalogów tylko katalogu bieżącego, oprócz sytuacji gdy:

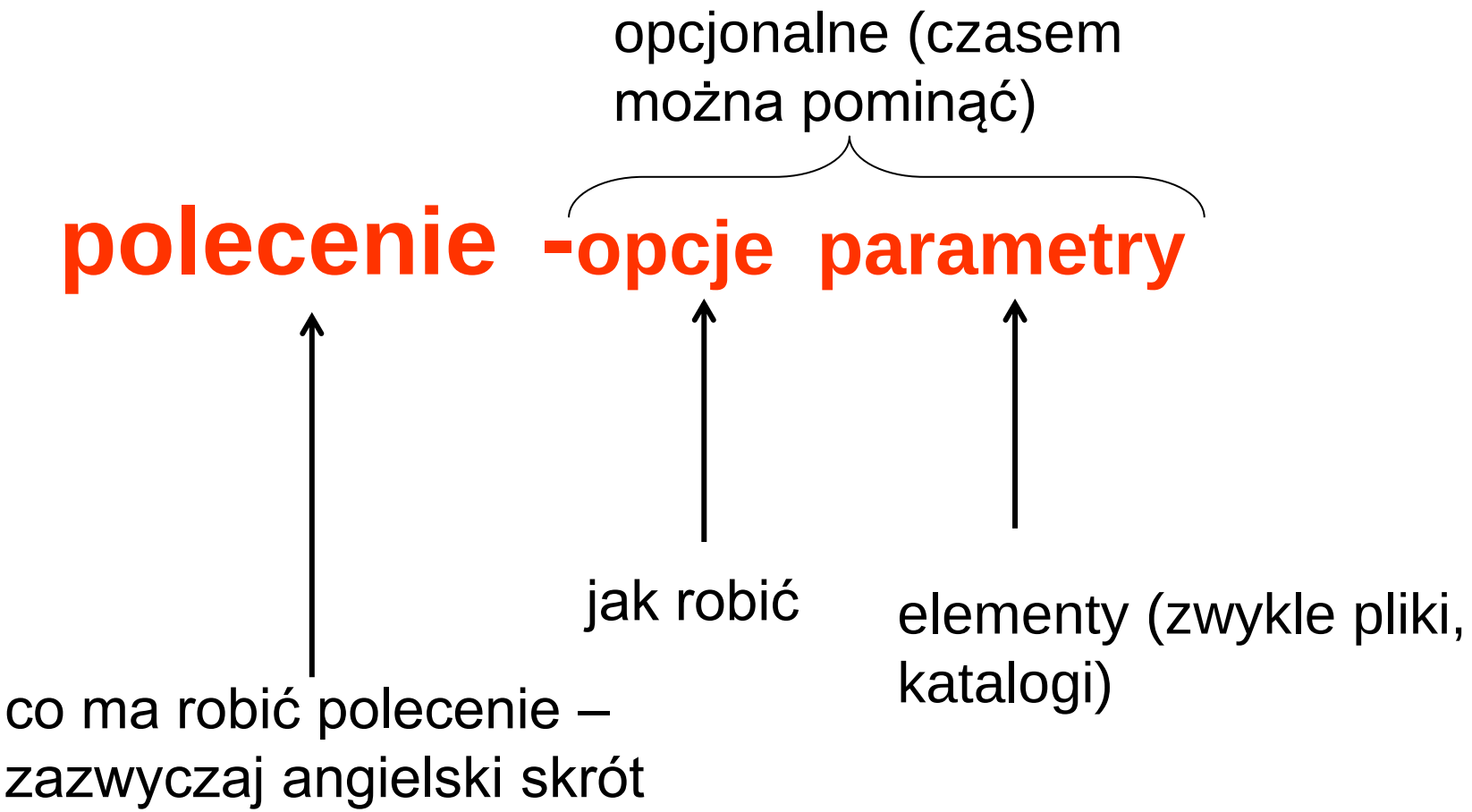
- podamy w poleceniu pełną ścieżkę dostępu
- plik wykonywalny (program) znajduje się w katalogu włączonym do tzw. ścieżki przeszukiwań

Ścieżka może być:

bezwzględna - począwszy od katalogu głównego /

względna - począwszy od katalogu bieżącego

Ogólna składnia polecenia powłoki:



Podstawowe polecenia

Pokazanie ścieżki bezwzględnej do katalogu bieżącego (w którym aktualnie pracuje użytkownik)

pwd

```
pwd
        /home/marek_s
```

Wyświetlenie spisu elementów katalogu bieżącego (pliki i podkatalogi)

`ls` spis samych nazw

`ls -l`

wyświetla pełną zawartość katalogu bieżącego – opcja **`-l`** pełne informacje o prawach, rozmiarze, dacie utworzenia, właścicielu

`ls –li` - opcja **`i`** – dodatkowo numer i-węzła

`ll` – krótki odpowiednik **`ls -l`** (nie we wszystkich systemach unixowych)

Można używać wzorców nazwy

Znaki specjalne wzorca

*	**zastępuje dowolny ciąg znaków**
?	**zastępuje jeden znak**

czyli..

`ls -l a*` wyświetl tylko pliki i katalogi (też ich zawartość) o nazwie na literę a

`ls -l p??` wyświetl tylko pliki i katalogi o nazwie na literę **p** i nazwie 3-znakowej

Nawigacja po strukturze katalogów

cd – *change directory*

cd ***ścieżka*** - zmiana katalogu bieżącego

Jeśli *ścieżka* rozpoczyna się od znaku ***/*** (rozpocznij od katalogu głównego) to jest to tzw. **ścieżka bezwzględna**, np.:

cd / - przejdź do katalogu głównego

cd /home - zmień katalog bieżący na *home*, który jest w katalogu głównym

cd /home/ala - zmień katalog bieżący na *ala*, który jest podrzędny do katalogu *home* w katalogu głównym

Ścieżka względna określa położenie katalogu (pliku) względem katalogu bieżącego, np.:

cd ..	- zmiana katalogu bieżącego na nadrzędny
cd	- zmiana katalogu bieżącego na domowy użytkownika (powrót do domowego) lub cd ~
cd ../KAT	- zmiana katalogu na inną „gałąź” („wyjdź wyżej” i „wejdź” do KAT) – droga w górę i w dół struktury
cd ../..	- przejdź 2 poziomy wyżej

Operacje na katalogach

ang. make remove move

Tworzenie i usuwanie katalogu

mkdir ***nazwa***	- tworzenie katalogu (podrzędnego jeśli nie podamy ścieżki)
rmdir ***nazwa***	- usunięcie katalogu (pustego!)

Usunięcie katalogu niepustego

rm –r ***katalog_podrz*** opcja –r (rekursywnie)

Zmiana nazwy katalogu

mv ***nazwa_stara nazwa_nowa***

Oczywiście jeśli podajemy same nazwy to dotyczy katalogu podrzędnego – inaczej wymagana ścieżka

Operacje na plikach

Tworzenie nowego pliku tekstowego (pustego!!) – bez treści

touch plik

Tworzenie nowego pliku z treścią

cat> plik

piszemy treść – można wiele wierszy

……

naciskamy CTRL+D (koniec!)

Usuwanie plików

rm nazwa - usunięcie pliku – jeśli jest on w katalogu bieżącym, inaczej trzeba podać ścieżkę

rm wzorzec - usunięcie plików według wzorca

Przykłady:

rm dokument1 usuwa plik w katalogu bieżącym

rm ../pismo45 usuwa plik w katalogu nadrzędnym

rm a?? usuwa pliki o nazwie na literę a i dwa dowolne znaki w nazwie (razem 3)

rm * usuwa wszystkie pliki w katalogu bieżącym

Kopiowanie plików - *copy*

cp źródło cel

Przykłady:

cp plik1 plik2 - **źródło i cel w tym samym katalogu bieżącym – koniecznie INNA NAZWA!**

cp ../plik1 plik2 - **źródło w katalogu nadrzędnymi, a cel w katalogu bieżącym - INNA NAZWA!**

cp ../plik1 . - **źródło w katalogu nadrzędnymi, a cel w katalogu bieżącym - TA SAMA NAZWA**

cp plik1 ./KAT/plik2 - **źródło w katalogu bieżącym a cel w podrzędnym do bieżącego katalogu KAT**

Można używać wzorców do kopiowania wielu plików

Kopiowanie katalogów z zawartością (nawet z podkatalogami)

cp -r *ścieżka/*KAT *ścieżka/*KAT2

Zmiana nazwy (przeniesienie) pliku - *move*

mv ***źródło cel***

Przykłady

unikalna nazwa
w katalogu

mv ***plik1 plik2*** **-** ***zmiana nazwy (pliku lub katalogu)***
(plik1 w katalogu bieżącym!)

mv ***plik1 katalog*** ***- przeniesienie do istniejącego katalogu podrzędnego***

istnieje!

istnieje! w katalogu
bieżącym

Jeśli chcemy kopiować (przenosić, zmieniać nazwy) pliki z innych katalogów niż bieżący – trzeba podać ścieżkę np:

../plik - plik jest w katalogu nadrzędnym

../katalog/plik - plik jest w katalogu "sąsiednim"

/plik – plik jest w katalogu głównym

/root/plik – plik jest w katalogu root podrzędnym do głównego

Jak nadmieniono wyżej przy kopiowaniu (przenoszeniu, usuwaniu) grupy plików można stosować wzorce nazw