



STRUKTURE PODATAKA I ALGORITMI 1

VEŽBE 1

Tijana Ristović
Aleksa Cerovina
Filip Radovanović
Đorđe Nedić

Strukture podataka i algoritmi 1		Maksimalan broj poena	Okvirni termini
	Prisustvo predavanjima	2	po rasporedu
	Prisustvo vežbama	2	po rasporedu
	I TEST	9	krajem marta
	II TEST	11	početak maja
	I KOLOKVIJUM	21	sredina aprila
	II KOLOKVIJUM	25	kraj maja
	POPRAVNI KOLOKVIJUM I TESTOVI	?	
ZAVRŠNI DEO ISPITA		30	

36 poena sa predispitnih obaveza uslov za završni deo ispita



PREDISMITNE OBAVEZE	70 poena
ZAVRŠNI DEO ISPITA	30 poena
UKUPNO	100 poena

BROJ POENA	OCENA
[0, 50]	5
[51,60]	6
[61, 70]	7
[71,80]	8
[81, 90]	9
[91,100]	10



Pomoć

man *komanda*

man ls

komanda --help

ls --help

pwd - koji je tekući direktorijum

ls - izlistava sadržaj tekućeg ili navedenog direktorijuma

ls

ls /etc nije isto što i ls etc

ls -la /etc

cd - promena aktivnog direktorijuma

cd .. prelazak u direktorijum jedan nivo iznad

cd /home

cd maja

cd ../..

cd /

cd ~ prelazak u home direktorijum logovanog usera



mkdir imedir - kreira direktorijum

rmdir imedir - briše direktorijum (prazan)

cp izvor odrediste kopiranje fajlova, direktorijuma itd

 cp /etc/p* . kopiranje p* iz /etc u tekući
direktorijum

mv izvor odrediste - premeštanje

rm ime - brisanje fajlova

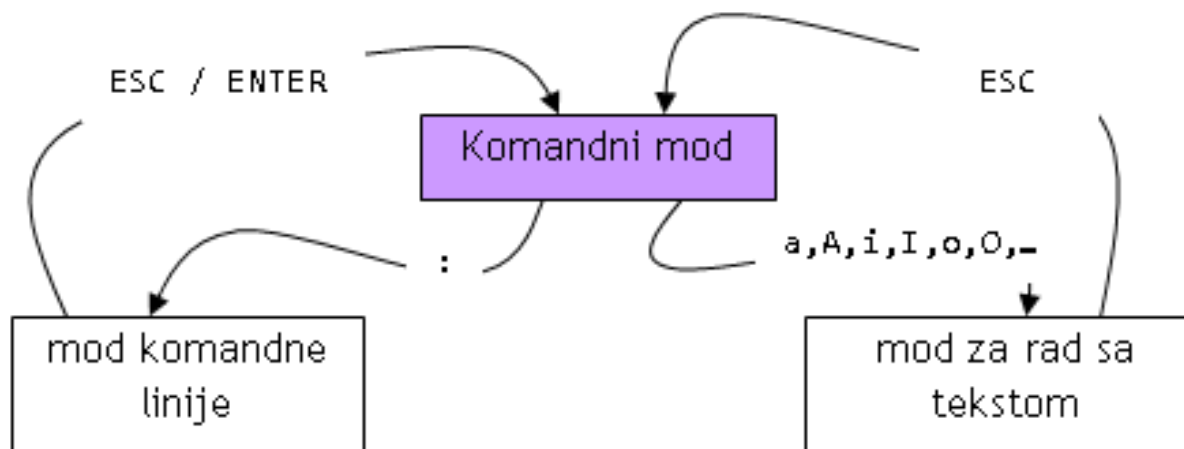
Napomena: Osnova editora vi je definisana početkom sedamdesetih, istovremeno sa početkom razvoja UNIX-a. Još onda je postavljen cilj da vi funkcioniše na raznim tipovima terminala od kojih većina nije imala ni preveliki ni premoćan skup kontrolnih sekvenci, kao ni standardizovan izgled tastature.

Modovi rada editora

Ekranski editor VI se može naći u jednom od tri moda rada:

- * **komandni** - svi znaci otkucani sa tastature se tretiraju kao komande,
- * **insert** - služi za unos teksta, tasteri imaju normalno značenje,
- * **mod komandne linije** - služi za unos dužih komandi / poziv shell komandi.

Nakon startovanja, editor ulazi u komandni mod. Prelazak u insert ili mod komandne linije je jedino moguć iz komandnog moda. Prelazak iz komandnog u insert mod se ostvaruje većim brojem komandi za dodavanje teksta ali se napuštanje insert moda i povratak u komandni obavlja uvek pritiskom na <Esc> taster. U mod zadnje linije je moguće preći jedino komandom ':'. Iz ovog moda se izlazi unošenjem željene komande i njenim izvršavanjem pritiskom na taster <Enter> ili <Esc> tasterom kada se mod zadnje linije napušta odmah.



Poziv editora

vi file - postavlja kursor u prvu liniju datoteke file.

vi + file - postavlja kursor na zadnju liniju datoteke file.

vi file1 file2 ... fileN - sukcesivno editovanje više datoteka. Po završetku editovanja jedne, sledeća se poziva sa :next ili skraćeno :n.

Izlaz iz editora

:wq - zapiši i izadi

:q! - izadi bez zapisivanja — bez obzira na učinjene izmene.

:w! - prepisi datoteku iako je 'Read only'

:x - isto što i :wq

Poziv shell komandi

:! komanda

:! gcc -o h hello.c

:! ./h

Pomoć

:help

:quit

Poziv drugih datoteka

:e file - u editor poziva drugu datoteku na editovanje. Nova datoteka briše iz bafera prethodni tekst.

:r file - učitava datoteku file i umeće je u tekst ispod tekuće linije.

Promenljive VI-a

Slično kao komandni interpreter, i VI poseduje neke ugrađene promenljive, koje ako su postavljene, menjaju ponašanje editora. Na sve promenljive ovog tipa se može uticati `:set` komandom. Na primer:

`:set nu` prebacuje VI u displej mod gde je, osim sadržaja datoteke, na početku prikazan i redni broj linija. Ovo može biti od koristi jer velik broj komandi koristi redne brojeve linija kao argumente.

`:set nonu`

`:set showmode` na kraju zadnje linije će se prikazivati trenutni mod rada editora (komandni ili insert).

`:set noshowmode`

`:set showmatch` editor će prikazivati uparenost zagrada `'(){}[]'` sto je vrlo korisno u razvoju programa.

`:set noshowmatch`

Komande za zapisivanje

- :i,jw file2** zapisuje tekst od i-te do j-te linije u file2
- :w file2** zapisuje celu datoteku pod drugim imenom file2
- :w>> file2** čitav sadržaj zapisuje na kraj datoteke file2

Pomeranje kursora

- ^** - pomeranje kursora na prvi neblanko znak tekuće linije
- \$** - pomeranje kursora na kraj tekuće linije
- <Enter>** - pomeranje kursora na početak naredne linije
- <Ctrl> f** - pomera ceo prozor unapred kroz datoteku
- <Ctrl> b** - pomera ceo prozor unazad kroz datoteku

UNDO i REPEAT

- u** - vraća datoteku na stanje pre izvršavanja poslednje komande
- .** - ponavlja poslednju komandu

Unos teksta

- a** unošenje teksta počev od pozicije prve udesno od pozicije kursora
- A** unošenje teksta na kraj tekuće linije
- i** unošenje teksta počev od tekuće pozicije kursora
- I** unošenje teksta pre prvog ne-blanko karaktera u liniji
- o** kreira novu liniju ispod tekuće i ulazi u 'insert' mod
- O** kreira novu liniju iznad tekuće i ulazi u 'insert' mod

Rad sa baferima

VI poseduje dve vrste bafera (i.e. clipboard): neimenovane i imenovane.

Neimenovanih ima 9 a referenciranje na njih se obavlja navođenjem njihovog rednog broja. Komande **dd** i **yy** menjaju sadržaj ovih bafera tako što se novi sadržaj uvek upisuje u prvi a pri tome se stari sadržaj bafera pomera iz prethodnog u naredni pri čemu se sadržaj poslednjeg (devetog) gubi.

Imenovanih bafera ima 26 a imena im se sastoje od slova engleske abecede. Mala i velika slova predstavljaju fizički iste bafere, ali se u radu razlikuju. Na primer, zapis u bafer **a** briše njegov prethodni sadržaj dok je zapis u bafer **A** zapis u isti bafer, ali se novi sadržaj dodaje prethodnom.



- x - briše znak ispred kursora
- X - briše znak iza kurzora
- dd - **briše tekuću liniju i smešta njen sadržaj u neimenovani bafer.**
- dw - **briše tekuću reč**
- :i,jd - **briše tekst od i-te do j-te linije**
- p - **štampa sadržaj prvog neimenovanog bafera ispod tekuće linije**
- P - **štampa sadržaj prvog neimenovanog bafera iznad tekuće linije**
- yy - **smešta tekuću liniju u neimenovani bafer (ne briše je)**
- "anyy pamti u baferu 'a' n linija
- "ap štampa sadržaj bafera a ispod tekuće linije.
- "np štampa sadržaj n-tog neimenovanog bafera ispod tekuće linije

Za Windows ovisnike:

p - paste

d – cut

y – copy

v – selekcija dela teksta, tačnije pokreće tzv. visual mod

Multipliciranje komandi

Instrukcije komandnog moda je moguće multiplicirati, jednostavno navođenjem željenog broja pre unosenja komande.

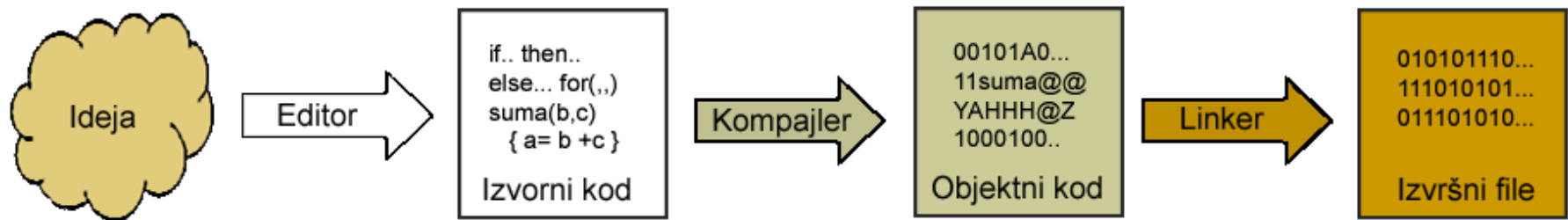
nkomanda

Na primer:

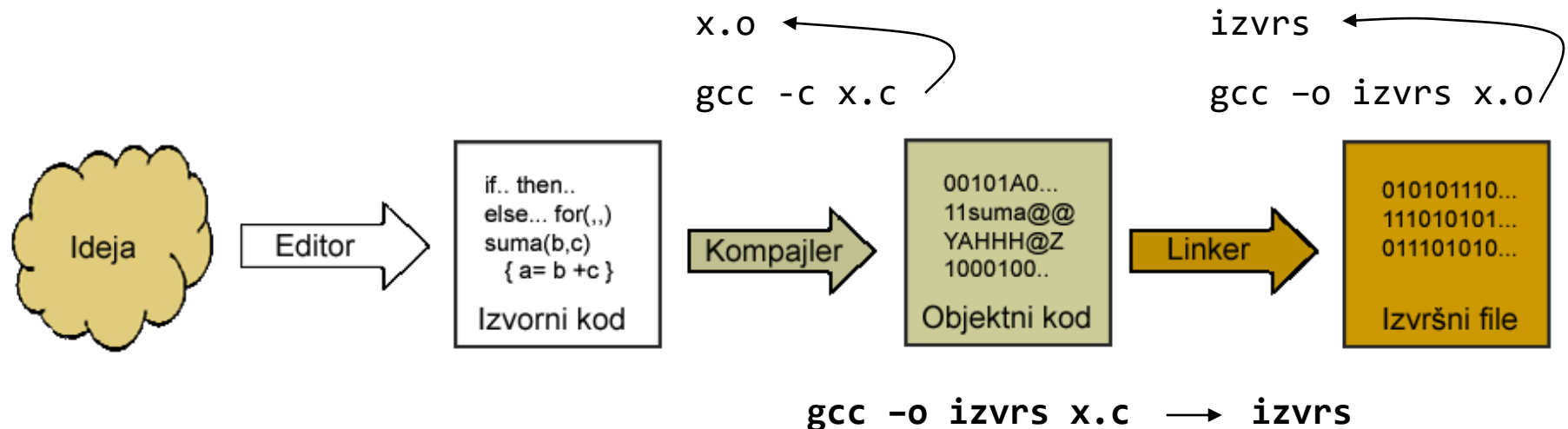
10x briše deset znakova desno od kursora

10dd briše 10 linija, i slično.

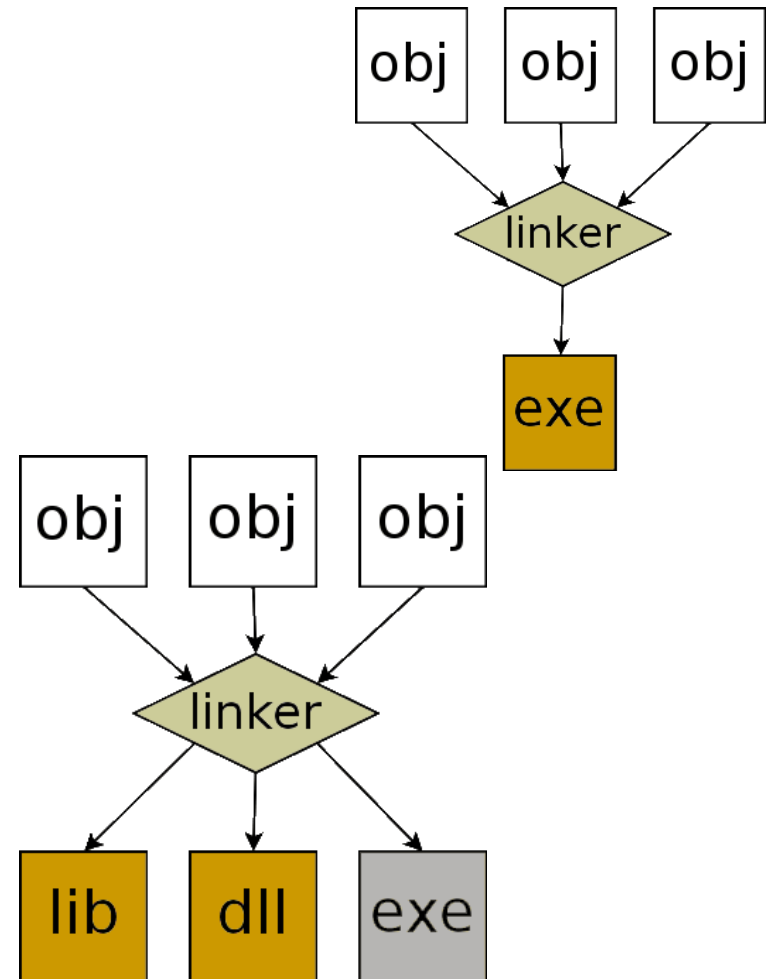
- **Izvorni programski kod** (engl. *source code*) je skup naredbi napisan u nekom od programskih jezika.
- **Objektni kod** (engl. *object code*) je mašinski kod generisan iz izvornog koda.
 - Objektni kod, iako je mašinski kod, ne može se izvršavati.
- **Izvršni fajl** (engl. *executable*) je fajl čiji sadržaj računar interpretira kao program.
 - Izvršni fajl se može pokretati samostalno, nezavisno od okruženja u kojem je razvijan. Sačinjen je od binarnih informacija (0 i 1).



- U najširem smislu, Compiler je alat koji prevodi tekst napisan u nekom od programskih jezika u drugi programski jezik.
- Praktična upotreba je prevođenje iz programskog jezika višeg u jezik nižeg nivoa, najčešće u simbolički mašinski jezik (često se naziva i assembler) ili mašinski jezik.



- Linker je alat koji od jedne ili više datoteka koje sadrže objektni kod kreira jedinstveni izvršni file ili biblioteku.
- **Biblioteke (eng. library) su skupovi potprograma.**
 - Statičke biblioteke (lib) se koriste u prevođenju i njihov kod se neposredno ugrađuje u izvršni program.
 - Dinamičke biblioteke (dll) se ne ugrađuju u izvršni program, ali moraju biti dostupne kada se program izvršava.



- Šta je rezultat sledećeh koda?

```
#include<stdio.h>
main()
{
    int x=4;
    float y=4.9;
    if((int)(y-x)) printf("%d", (int)(y+x));
    printf("%.2f\n", (float)(x+2)/x);
}
```

- Uneti jednocifren ceo broj **k** i ceo broj **n** a zatim **n** realnih brojeva. Koliko brojeva od unetih **n** brojeva ima prvu decimalnu sifru jednaku broju **k**?

```
#include<stdio.h>
main()
{
    int k,n,i,c,br;
    float x;

    scanf("%d%d",&k,&n);
    br=0;
    for(i=1;i<=n;i++)
    {
        scanf("%f",&x);
        c=(int)(10*x)%10;
        if(c==k) br++;
    }
    printf("%d\n",br);
}
```