

VEŽBA 4

Pregled

Prevođenje programa napisanih u jeziku C izvodi se u dva koraka. Prvi korak je prevođenje programskog modula iz izvornog koda u objektni, a uz korišćenje kompajlerskog sistema *gcc* postiže se sledećom komandom:

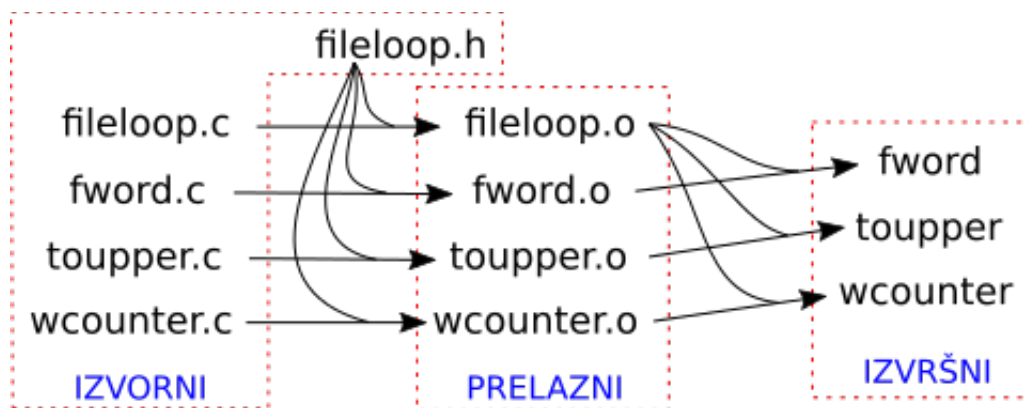
```
gcc -Wall -O1 -c c_prog.c
```

Opcija *-Wall* uključuje prikaz svih mogućih poruka upozorenja, a *-O1* uključuje prvi stepen optimizacije generisanog mašinskog programa. Prekidač *-c* nalaže kompajleru da ne obavlja linkovanje modula i generisanje izvršnog programa, nego da generiše samo objektni kod modula koji će biti zapisan u fajl pod istim imenom kao i tekst programskog modula u jeziku C (ekstenzija *.c*), a ekstenzija će podrazumevano biti *.o*.

Objektni moduli se ne mogu izvršiti na sistemu (ne mogu se pozvati iz shell-a) i najčešće sadrže samo deo funkcionalnosti kompletnog programa. Oni su programski moduli nepotpuno prevedeni iz programskog modula napisanog u jeziku C. Izvršni fajl se dobija povezivanjem svih potrebnih objektnih modula i odgovarajućih sistemskih biblioteka. Pri korišćenju *gcc* kompajlerskog sistema linkovanje se obavlja komandom:

```
gcc -o izvrsni_fajl modul_1.o modul_2.o
```

Prekidačem *-o* označeno je željeno ime generisanog izvršnog fajla.



Zadaci

1. Napisati *makefile* koji opisuje zavisnosti u paketu programa sa gornje slike. Strelice pokazuju prema fajlu koji je određište (*target*), a strelice polaze od fajlova od kojih određište zavisu. *Makefile* treba da ima i targete *all* i *clean*. Prvim se postiže kompajliranje sva tri izvršna programa i on treba da se navede kao prvi u nizu targeta, dok drugi treba da obezbedi brisanje svih objektnih fajlova paketa.

Preuzeti programske module `fileloop.c` i `fword.c`. Testirati *makefile* zadavanjem komande `make fword` čime se traži generisanje programa `fword`, bez pokušaja kompajliranja

ostalih modula (koji još nisu napisani).

Nakon uspešnog kompajliranja programa *fword* potrebno je testirati ga. Poziva se kao komanda sa jednim argumentom, a taj jedan argumet je reč koja se traži u zatom tekstu. Jedini način da se zada tekst za pretraživanje jeste da se on dovede kao standardni ulaz komande, a to se najlakše postiže preusmeravanjem ulaza.

Primer:

```
fword Zastava < z101.txt
```

Ovom komandom pretražiće se tekstualni fajl *z101.txt* da bi se utvrdilo sadrži li barem na jednom mestu rec „Zastava“. Komanda razlikuje velika i mala slova, pa o tome treba voditi računa. Komanda ne ispisuje nikakav rezultat na standardni izlaz, rezultat pretrage treba tražiti preko vraćenog statusa komande koji je jednak *0* ako je reč pronađena ili *1* ako nije pronađena. Rezultat se, dakle može proveriti pomoću *if* konstrukcije iz shell skripta ili štampanjem specijalne promenljive *\$?* neposredno nakon poziva komande.

Primer:

```
echo $?
```

2. Napisati programski modul u jeziku C za generisanje komande *toupper*. Generisati izvršni fajl zadavanjem komande *make toupper*. Testirati ga.

Specifikacija programa *toupper*:

Osnovna svrha programa je da ulazni tekst pretvori u isti taj tekst, ali napisan isključivo velikim slovima. Dakle, svaki karakter *ch* treba proveriti nalazi li se u opsegu *'a' <= ch <= 'z'*, a ako se nalazi treba od njegovog ASCII koda oduzeti vrednost *('a' - 'A')*. Sam program ne zahteva nikakve parametre, a svoj izlaz daje preko standardnog izlaza.

Program *toupper* treba da koristi funkciju *file_loop* koju definiše *framework* zadat u izvornim fajlovima *fileloop.c* i *fileloop.h* na isti način kako je to učinjeno u primeru *fword*. Za ispis karaktera na standardni izlaz koristiti funkciju *putchar(c)* koja ispisuje jedan karakter prosleđen preko svog jedinog argumenta.

Primer:

```
toupper < z750.txt
```

3. Napisati programski modul u jeziku C za generisanje komande *wcounter*. Generisati izvršni fajl zadavanjem komande *make wcounter*. Testirati ga.

Specifikacija programa *wcounter*:

Osnovna svrha programa je brojanje reči u tekstualnom fajlu. Kao reč se definiše svaka skupina karaktera različitih od praznog mesta *' '* ili karaktera za novi red *'\n'*. Program ne zahteva parametre, ulazni fajl čita preko standardnog ulaza, a rezultat brojanja ispisuje u vidu broja bez karaktera za novi red na kraju na standardnom izlazu.

Program *wcounter* (kao i *toupper*) treba da koristi funkciju *file_loop* koju definiše *framework* zadat u izvornim fajlovima *fileloop.c* i *fileloop.h* na isti način kako je to

učinjeno u primeru *fword*. Za ispis formatiranog decimalnog broja na standardni izlaz koristiti funkciju `printf` u obliku `printf("%d", wordcount)` koja ispisuje vrednost promenljive *wordcount* tipa *int* u decimalnom obliku bez prelaska na novi red.

Primer:

```
wcounter < Florida.txt
104
```

Iz definicije reči sledi da se njihov broj može utvrditi tako što će se za svaki karakter fajla proveriti da li je prazno mesto ili karakter za novi red, a ako jeste brojač reči treba uvećati za jedan. Algoritam se dalje može unaprediti tako da prepoznađe sukcesivna prazna mesta ili karaktere za novi red u nizu, a za sve njih brojač uveća samo jednom.

4. Zadati komandu *make clean* da bi se uklonili svi objektni fajlovi, nakon čega se može testirati ispravnost prevođenja celokupnog sistema komandi zadavanjem samo komande *make*. Testirati sistem i promenom samo pojedinih izvornih fajlova i utvrditi ispravnost sleda komandi koje *make* automatski generiše za prevođenje.
5. Napisati shell skript koji omogućava da se izborom iz menija testiraju sve funkcionalnosti komandi napisanih u jeziku C. Meni treba da bude implementiran shell konstrukcijom *select* i da ima sledeće opcije:

```
1) Zadavanje fajla za obradu      4) Brojanje reci
2) Ispis u izvornom obliku        5) Traženje reci
3) Konverzija u velika slova     6) Izlaz
#?
```

Fajl koji služi kao standardni ulaz za sve opcije (osim 1 i 6) zadaje se opcijom 1, a opcije od 2 do 5 proveravaju postojanje zadatog fajla, a ako on postoji pozivaju odgovarajuću komandu. Ako fajl ne postoji ispisuje se odgovarajuće upozorenje. Ukoliko su potrebni dodatni parametri oni se traže od korisnika preko terminala pomoću komande *read*. Opcija 6 omogućava izlazak iz menija i završetak rada skripta.

Primer:

```
#? 1*
Unesite ime fajla : Florida.txt
Prihvaceno! Dalje ce se obradjivati fajl pod imenom
Florida.txt
```

```
#? 4
Brojanje reci : Florida.txt
Fajl Florida.txt sadrzi otprilike 104 reci.
```

```
#? 6
Pozdrav!
```

* Zasivljena polja označavaju interaktivno unešen tekst od strane korisnika