Makefile za Početnike

Zašto Makefile?

Makefile je program za kompilaciju koji vodi računa da ne rekompajluju delovi koda koji nisu menjani od prošle kompilacije. Odnosno koristi se za vođenje računa o ažurnosti izvršne datoteke vašeg projekta (da vam svi delovi projekta od koje će se napraviti izvršna datoteka budu up-to-date), odnosno ako se nešto promeni u projektu, da se rekompajluju samo delovi projekta koji su zavisni od promenjenog dela.

Dakle, do sada ste vrlo verotano koristili jednostavne komande za kompilaciju i linkovanje:

```
gcc -Wall -Wextra -g -o Milivoje Milivoje.c -std=c99
```

Što je realno već malo predugačka komanda za pisanje od nule kada se piše više od `0` puta. Doduše, možda znate touch typing, možda želite da assertuje svoj dominance pa pišete ručno svaki put. E pa onda mi demonstrirajte svoj big dick energy na sledećoj komandi:

```
flex lexer.lex|g++ -Wall -Wextra -std=c++14 -c -o lex.yy.o lex.yy.c|g++ -Wall -
Wextra -std=c++14 -c -o lex.yy.o lex.yy.c|g++ -Wall -Wextra -std=c++14 -c -o
parser.tab.o parser.tab.cpp|g++ -Wall -Wextra -std=c++14 -c -o the_set.o
the_set.cpp|g++ -o sets the_set.o lex.yy.o parser.tab.o
```

Uvod

Tu dolazi u priču Makefile, sa kojim možemo da radimo nekoliko stvari:

1. Da definišemo hijerarhiju zavisnosti datoteka u projektu
2. Da definišemo takozvane .PHONY funkcije, koje koristimo nezavisno od hijerarhije
3. Da definišemo `akcije` odnosno komande koje Makefile treba da izvrši u slučaju da se neka zavisnost naruši

> Setup Makefile-a:
>
> 1. Napraviti fajl u `direktorijumu u kome se nalazi projekat` sa imenom Makefile
> 2. Ispisati željeni Makefile kod koji će biti objašnjen ispod u `Makefile datoteku`
> 3. Pozicionirati se u terminalu na direktorijum projekta i ukucati `make`, što pokreće ažuriranje projekta
> 4. Ako želite da pokrenete neku od `.PHONY` funkcija, npr `clean`, u terminalu ukucati `make clean`.
> 5. Isto tako se može izvršiti kompilacija samo konkretnog fajla, odnosno može se izvršiti **recept** vezan za konkretnu datoteku, npr `make naš_program`ile treba da izvrši u slučaju da se neka zavisnost naruši

Primer

```
nas_program: nas_kod1.c
    gcc nas_kod1.c -o nas_program
# komentari se pišu sa tarabom na početku
```

Ovo znači da je `naš_program` zavistan od `nas_kod1.c`, što znači da ako se promeni `naš_kod1.c` na bilo koji način, izvršiće se komanda koju smo definisali ispod.

```
nas_program:                    nas_kod1.c
# cilj (ime programa)            # lista zavisnosti (za sada ima samo jedan
element()
    gcc nas_kod1.c -o nas_program
    # komanda koja se izvršava ako se narusi zavisnost
```

Ovo isto možemo da radimo za više izvornih datoteka:

```
nas_program: ovo.o je.o bas.o dosta.o datoteka.o
    gcc ovo.o je.o bas.o dosta.o datoteka.o -o nas_program
ovo.o: ovo.c
    gcc -c -o ovo.o ovo.c
je.o: je.c
    gcc -c -o je.o je.c
bas.o: bas.c
    gcc -c -o bas.o bas.c
dosta.o: dosta.c
    gcc -c -o dosta.o dosta.c
datoteka.o: datoteka.c
    gcc -c -o datoteka.o datoteka.c
```

Da se podsetimo kod kompajolavanja više datoteka, mora se prvo napraviti objektni modul od svake izvorne datoteke posebno sa `-c` a onda tek da se svi objektni moduli linkuju sa skroz gornjoj komandom.

Dakle, ako se promeni `je.c`, izvršiće se komanda `gcc -c -o je.o je.c` i time će se ažurirati `je.o`, sto onda znači da se i on menja, što znači da se mora ažurirati i `naš_program`, što znači da se izvršava komanda `gcc ovo.o je.o baš.o dosta.o datoteka.o -o nas_program` i time je ažuriran ceo projekat.

Gorepomenute PHONY funkcije su poprilično jednostavne i mogu se na sledeći način koristiti:

```
.PHONY: clean

clean:
    rm *.o
```

Ovo prosto rečeno znači da postoji funkcija koja nije zavisna od datoteke koja se zove `clean` koja da bi se ispunila ažurnost (aka uvek kada se eksplicitno pokrene), mora da izvrši komandu `rm *.o` . Ovakve funkcije se pokreću kucanjem `make ime_funckije` odnosno u ovom slučaju `make clean`.

Korisne funckije Makefile-a

```
GOAL = nas_program
COMP = gcc

GOAL: ovo.o je.o bas.o dosta.o datoteka.o
    $(COMP) $^ -o $@
ovo.o: ovo.c
    $(COMP) -c -o $@ $<
je.o: je.c
    $(COMP) -c -o $@ $<
bas.o: bas.c
    $(COMP) -c -o $@ $<
dosta.o: dosta.c
    $(COMP) -c -o $@ $<
datoteka.o: datoteka.c
    $(COMP) -c -o $@ $<
```

Promenljive se definišu sintaksom: `ime_promenljive = vrednost` i dereferenciraju se sa dolar znakom: `$(ime_promenljive)`, u ovom slučaju smo ih koristili da skratimo kucanje, a i da bismo u slučaju da promenimo ime programa ili kompajlera ne moramo da kucamo sve od Kulina Bana.

Kao što se može videti, većina funkcionalnosti Makefile-a je vezano za $ simbol, ostatak funckija koje koristimo ovde su:

1. `$^`- makro koji vraća celu listu zavisnosti trenutne veze, u ovom slučaju

```
   $^ = ovo.o je.o bas.o dosta.o datoteka.o
```

2. `$<` - makro koji vraća najlevlji član liste zavisnosti
3. `$@` - makro koji vraća cilj u trenutnoj vezi, npr u `ovo.o: ovo.c` cilj je `ovo.o`

Poslednja korisna stvar, ako imamo više nezavisnih programa koje želimo da napravimo odjednom, definišemo `all` zavisnost, koja u suštini slično phony funkcijama se izvrši samo kada je eksplicitno pozovemo:

```
all: program1 program2 program3
```

Gde je koristimo tako što napišemo u terminalu `make all`

> Pro Tip:
>
> Ako pokušate da dodelite jednu promenjlivu drugoj u Makefile-u, doći će do beskonačne rekurzije. Ovo se popravlja tako što se umesto `=` koristi `:=`

Još Neke Korisne Funckije Makefile-a

Kao što se može videti, čim se pokrene bilo koja komanda u Makefile-u, ona se i ispiše u terminalu kao da ste je sami prokrenuli, ako ne želite da vam terminal eksplicitno ispisuje komande koje ste izvršili, možete da pišete `@` na početku naredbe, npr:

```
speak:
	@echo "RG je najjači predmet ikada!"
```

Tako da sada kada budemo pokrenuli Makefile, terminal će izgledati ovako:

```
$ make
RG je najjači predmet ikada!
```

Zaključak

Po meni je ovo sve što je potrebno za uvod u Makefile, ako vi drukčije mislite ili želite sami da editujete ovaj priručnik, pošaljite mi mail sa subjectom: `Makefile MD Tutorial` .

Autor:

Vladimir Batoćanin

email: `vladimir@wings.rs`