

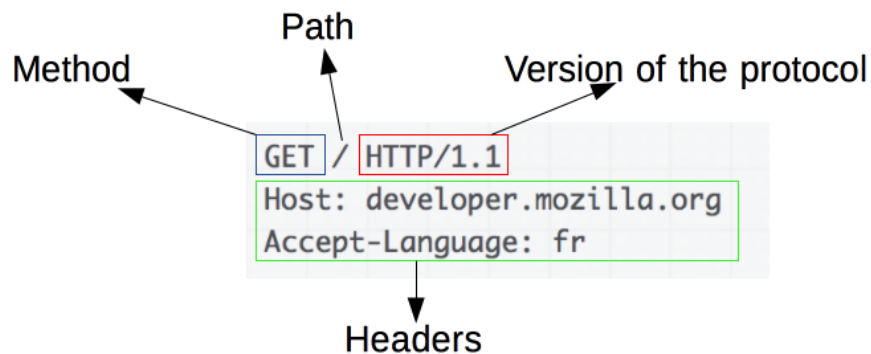
Zadaća 5 iz predmeta Dizajn kompajlera

Problem

Koristeći alate flex i bison potrebno je napisati parser za poruke HTTP protokola.

Problem 1

Primjer HTTP upita slijedi:



Upiti su sačinjeni od sljedećih komponenata:

- HTTP metod, koji je jedan od konačno mnogo identifikatora, kao što su GET, HEAD, POST, PUT, DELETE, OPTIONS (nije potrebno razmatrati druge).
- Putanja do resursa (URI) koji se traži. Dobije se tako što od URL-a skinemo početne elemente (http, domenu i port), a ostavimo krajnji dio, putanju.
- Verzije HTTP protokola,
- Opciona lista headera.

Headeri su dati u formatu field-name : field-value i rastavljeni su novim redom. Field-name možete smatrati identifikatorom. Field value je neterminal koji može

biti string ili broj.

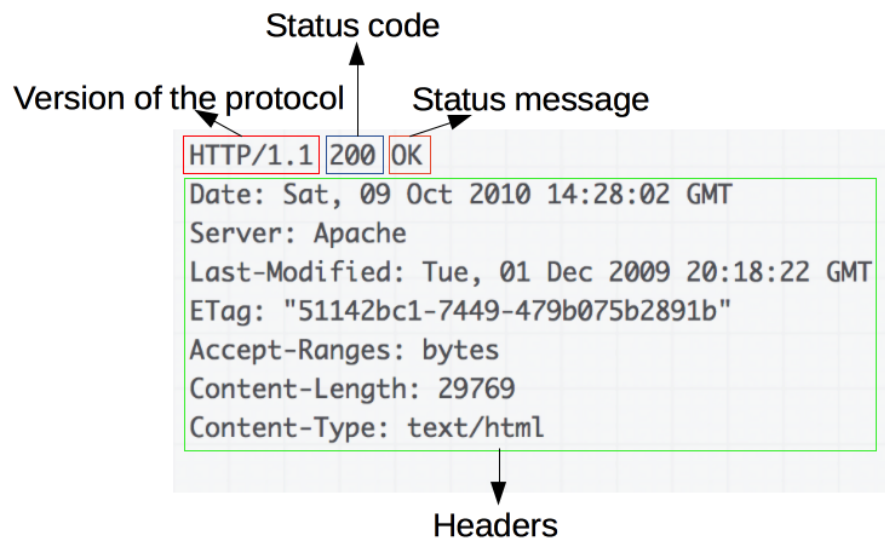
Program treba primiti tekst na standardni ulaz i u slučaju validnog ulaza ispisati korisniku metod i URI koji se tražio u primjeru. Za ulaz kao na slici iznad ispis bi bio:

GET /

U slučaju pogreške treba ispisati “Poruka nije tačna”.

Problem 2

Napisati parser HTTP odgovora, koji je u formatu kao na slici:



Odgovor je sačinjen od sljedećih elemenata: - Verzija HTTP protokola koju prate, - Statusni kod koji indicira da li je upit bio uspješan, - Statusna poruka, koja je kratki opis status koda, - HTTP zaglavlja, nalik onim u upitu.

Program treba primiti tekst na standardni ulaz i u slučaju validnog ulaza ispisati korisniku statusni kod i poruku. Za ulaz kao na slici iznad ispis bi bio:

200 OK

U slučaju pogreške treba ispisati “Poruka nije tačna”.

Par napomena, koje se odnose na oba zadatka:

- U primjeru na slici iznad, vidimo da pojedini HTTP headeri mogu imati datum kao vrijednost. Smatramo za ovaj zadatak da se takav ulaz neće

pojaviti te nije potrebno praviti lekser datuma,

- Ako je string sačinjen od više riječi, možete smatrati da će biti unutar navodnika, tako da parsiranje bude lakše,
- Nije potrebno parsirati *body* dio upita,
- Ukoliko naidete na poteškoće, možete smatrati field-name i status-message stringom, mada bi bilo dobro da oni imaju vlastite tokene.

Dodatna objašnjenja možete naći na stranici [<https://developer.mozilla.org/en-US/docs/Web/HTTP/Overview>]

Predaja zadaće

Za svaki problem potrebno je napraviti zaseban direktorij sa imenima problem1, problem2, i td. Unutar direktorija se nalaze sve potrebne datoteke za kompajliranje programa. Svaki problem mora imati svoj Makefile koji će komandom make kompajlirati program i napraviti jedan izvršni file koji će nositi ime kao i problem, dakle problem1, problem2 i td. Na primjer, naredna sekvenca shell komandi treba biti validna:

```
cd problem1
make
./problem1 < input_fajl
Ulaz je validan.
```

Program će koristit standardni ulaz i izlaz. Primjer Makefilea koji prepoznaje flex i bison možete pronaći u laboratorijama.