Lex Programlama Aracı Nedir, Ne İçin Kullanılır

Lex (Lexical Analyzer Generator), bir dilin leksik analizini gerçekleştirmek üzere kullanılan bir araçtır. Lex, bir dilin sözdizimi yapısını incelemek yerine, kaynak kodun içindeki belirli desenleri tanımak ve bu desenlere uyan kısımları işlemek için kullanılır. Lex programları genellikle bir giriş metni içinde belirli desenleri bulma ve bu desenlere göre işlemler gerçekleştirme amacına hizmet eder.

Lex aracının temel kullanım alanları şunlardır:

1. **Leksik Analiz (Tokenization):** Lex, genellikle bir dilin leksik analizini gerçekleştirmek için kullanılır. Kaynak kodu analiz ederken, belirli desenlere uyan kelimeleri veya sembollerin tokenlerini belirler. Bu tokenler daha sonra dilin sentaks analizi için kullanılabilir.
2. **Dilin Özelliklerini Tanımlama:** Lex programları, bir dilin özel belirli yapılarını tanımlamak için kullanılır. Örneğin, belirli bir dilin anahtar kelimelerini, sembollerini veya diğer özel yapılarını tanımlayabilir.
3. **Dil Çevirici (Compiler) Aşamalarında Kullanım:** Lex ve Yacc (Yet Another Compiler Compiler) gibi araçlar bir araya gelerek dil çevirici aşamalarını (lexical analysis, syntax analysis, semantic analysis) gerçekleştirebilir. Lex, giriş metnini analiz ederek dilin leksik yapısını belirler, Yacc ise dilin sentaks yapısını belirler.
4. **Metin İşleme:** Lex programları genellikle metin işleme görevleri için de kullanılır. Belirli desenlere uyan metinleri bulma, değiştirme veya filtreleme gibi işlemleri gerçekleştirebilirler.

Örneğin, bir dilin anahtar kelimelerini veya belirli desenlere uyan ifadeleri bulmak için Lex kullanabilirsiniz. Aşağıdaki örnek, bir metindeki sesli ve ünsüz harf sayılarını bulan bir Lex programının örneğidir.

Ubuntu İçin Lex Compiler kurulumu : Terminali açıp sırasıyla şu komutları girin;

sudo apt-get update

sudo apt-get install flex

sudo apt-get install byacc

sudo apt-get install bison

sudo apt-get install bison++

masa üstüne içinde çalışan bir program bulunan bir lex (“.l “ uzantılı olacak) dosyası oluşturun.

ardından terminali tekrar açın

cd Desktop dedikten sonra ls komutuyla oluşturdugunuz dosyayı görün.

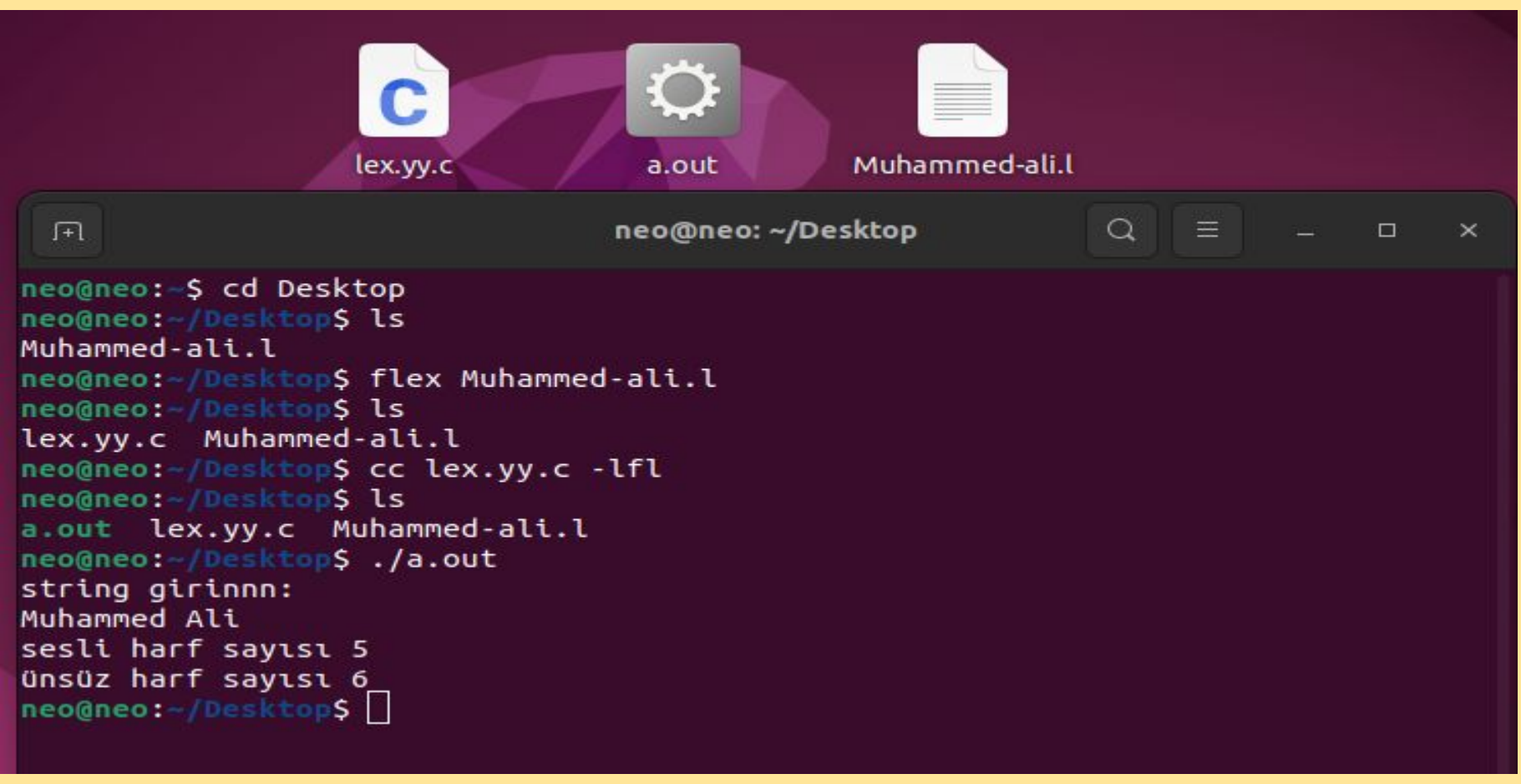
flex (dosya adı)

cc lex.yy.c -lfl

./a.out komutlarını çalıştırdıktan sonra programın istediği bilgileri girin ve enter'a basın

son olarak ctrl+d tuşlarıyla programın sonucunu görebilirsiniz.

Not= “lex.yy.c” ve “a.out” dosyaları siz terminalde kodları yazarken otomatik olarak masaüstüne oluşacaktır.



Programın Açıklanması;

1. `%{ ... %}` bloğu: Bu blok, C dilinde yazılmış kodu içerir ve genellikle global değişkenleri tanımlar. Bu örnekte `v` ve `c` isimli iki tane integer değişken tanımlanmış.
2. `%%` bloğu: Bu blok, Lex kurallarını içerir. Her bir kural, bir deseni ve bu desen eşleşirse ne yapılacağını belirtir.
   - `[ \t\n]+ ;` ifadesi: Bu ifade, boşluk, sekme veya yeni satır karakterlerini ignore eder.
   - `[aeiouAEIOU] {v++;}` ifadesi: Bu ifade, herhangi bir sesli harf bulunduğunda `v` değişkenini artırır.
   - `[^aeiouAEIOU] {c++;}` ifadesi: Bu ifade, herhangi bir sesli harf olmayan harf bulunduğunda `c` değişkenini artırır.
3. `int main()` fonksiyonu: Bu fonksiyon, programın başladığı yerdir. Kullanıcıdan bir metin girmesini ister, ardından `yylex()` fonksiyonunu çağırarak Lex analizini başlatır.
4. `int yywrap()` fonksiyonu: Bu fonksiyon, Lex analizini sonlandırmak için kullanılır. Bu örnekte her zaman 1 değerini döndürür, bu da analizin sonlandığını belirtir.

Bu program, girilen metindeki sesli ve ünsüz harf sayılarını hesaplar ve ekrana yazdırır. C dilinde yazıldığı için derlenerek çalıştırılabilir. Kullanıcıdan metin girmesini ister, ardından sonuçları ekrana yazdırır.

Yacc Programlama Aracı Nedir, Ne İçin Kullanılır?

Yacc (Yet Another Compiler Compiler), bir derleyici üretim aracıdır. Yacc, genellikle bir dilin sentaks analizini gerçekleştirmek ve dilin gramer kurallarına göre ifadeleri çözmek için kullanılır. Yacc, Lex (Leksik Analiz Üreteci) gibi araçlarla bir araya gelerek dil çevirici (compiler) aşamalarını tamamlar.

Yacc'ın temel kullanım alanları şunlardır:

1. **Sentaks Analizi (Syntax Analysis):** Yacc, bir dilin sentaks analizini gerçekleştirmek için kullanılır. Sentaks analizi, bir dilin gramer kurallarına uygunluk kontrolünü yapar ve dilin yapısal özelliklerini belirler.
2. **Dil Çevirici (Compiler) Aşamalarında Kullanım:** Lex ve Yacc gibi araçlar bir araya gelerek bir dilin çevirici aşamalarını tamamlar. Lex, leksik analizi (tokenization) gerçekleştirirken, Yacc sentaks analizini gerçekleştirir. Bu aşamalar, dilin kodunu anlamlandırma, hata kontrolü, semantik analiz ve son olarak makine kodu üretme gibi işlemleri içerir.
3. **Dilin Gramer Kurallarını Belirleme:** Yacc, bir dilin gramer kurallarını tanımlamak için kullanılır. Bu kurallar, dilin doğru ve tutarlı bir şekilde yazılmasını sağlar. Özellikle, ifadelerin nasıl bir araya getirileceğini, kontrol yapılarını ve fonksiyon çağrılarını tanımlayan kuralları içerir.
4. **Derleyici Tasarımı ve Geliştirmesi:** Yacc, yeni bir programlama dilinin derleyici (compiler) tasarımında ve geliştirmesinde kullanılır. Bir dilin sentaks analizini otomatikleştirerek, derleyici geliştiricilerin daha hızlı ve daha güvenilir derleyiciler oluşturmalarına olanak tanır.

Özetle, Yacc, bir dilin sentaks analizi ve gramer kurallarının belirlenmesi için kullanılan bir araçtır ve genellikle derleyici geliştirme süreçlerinde önemli bir rol oynar.

Ubuntu İçin yacc Compiler kurulumu : Terminali açıp sırasıyla şu komutları girin;

sudo apt-get update

sudo apt-get install flex

sudo apt-get install byacc

sudo apt-get install bison

sudo apt-get install bison++

masa üstüne içinde çalışan bir programlar bulunan lex ve yac dosyaları oluşturun.(“.l“.y “ uzantılı olacak)

ardından terminali tekrar açın

cd Desktop dedikten sonra ls komutuyla oluşturduğunuz dosyayı görün.

lex (dosya adı).l

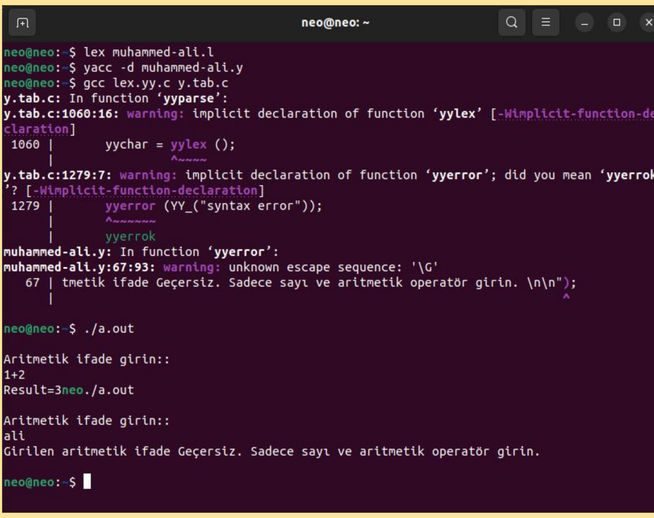
yacc -d(dosya adı).y

gcc lex.yy.c y.tab.c

./a.out komutlarını çalıştırdıktan sonra programın istediği bilgileri girin ve enter'a basın

Not= “lex.yy.c” , “a.out”, “y.tab.h” , “y.tab.c” dosyaları siz terminalde kodları yazarken otomatik olarak masaüstüne oluşacaktır.

Programın çıktısı



Programın Açıklanması;

Bu program, bir aritmetik ifade dilini analiz eden bir derleyici tasarlamak için kullanılan bir Lex (Flex) ve Yacc (Bison) kombinasyonunu içerir. Program, basit bir aritmetik ifadeyi (toplama, çıkarma, çarpma, bölme gibi) ve karşılaştırma operatörlerini (örneğin, `>=`, `<=`, `!=`, `==` gibi) değerlendirebilir.

Lex (Flex) Kodu Açıklaması:

1. `%{` ve `%}`: Bu blok içindeki kod, Lex programının C dilindeki kodunu içerir. Bu kısım, derlenmiş programa eklenen C dilindeki ön tanımlı kodlardır.
2. `%token NAME NUMBER`: Bu ifadeler, Flex programında kullanılacak sembollerin (token'ların) adlarını tanımlar.
3. `%left GE LE EQ NE EE '<' '>'`: Bu ifadeler, operatörlerin öncelik seviyelerini belirler.
4. `%%`: Bu işaret, Flex ve Bison kodları arasındaki bölümü ayırır.
5. Kurallar Bölümü (`ArithmeticExpression` ve `E` kuralları): Bu bölüm, Flex tarafından üretilen token'ları ve bu token'ların nasıl işleneceğini tanımlar. Her bir kural, bir sembolü tanımlar ve semboller arasındaki ilişkiyi belirler.

Yacc (Bison) Kodu Açıklaması:

1. `%{` ve `%}`: Bu blok içindeki kod, Bison programının C dilindeki kodunu içerir. Bu kısım, derlenmiş programa eklenen C dilindeki ön tanımlı kodlardır.
2. `%token NAME NUMBER`: Bu ifadeler, Bison programında kullanılacak sembollerin (token'ların) adlarını tanımlar.
3. `%left GE LE EQ NE EE '<' '>'`: Bu ifadeler, operatörlerin öncelik seviyelerini belirler.
4. `%nonassoc UMINUS`: Bu ifade, tekil işlem (negative işlemi gibi) operatörlerin önceliğini belirler.
5. `%%`: Bu işaret, Flex ve Bison kodları arasındaki bölümü ayırır.
6. Kurallar Bölümü (`ArithmeticExpression` ve `E` kuralları): Bu bölüm, Flex tarafından üretilen token'ları ve bu token'ların nasıl işleneceğini tanımlar. Her bir kural, bir sembolü tanımlar ve semboller arasındaki ilişkiyi belirler.
7. `int main()`, `int yyerror()`, `int yywrap()`: Bu bölüm, programın başlatılması, hataların işlenmesi ve lex analizinin sona erdirilmesi için gerekli olan C dilindeki kodları içerir.