

الجلسة الثانية

البدء ببناء مترجم بسيط يقوم بعملية Scanning فقط

الهدف من الجلسة

- التعريف بكيفية كتابة ملف وصف Scanner Description
- كتابة ملف Scanner Description لعملية مسح كود مصدري والتحقق من مطابقته للقوالب.

مستلزمات الجلسة

- حاسب بمواصفات دنيا RAM: 1 GB, CPU: 1.6 GHz, Windows 10 OS 64 bit
- Turbo C++/ DevC++
- LEX & BISON tools

الخلاصة والنتائج:

يفترض عند نهاية الجلسة:

تمكن الطالب من فهم بنية ملف وصف الماسح Scanner Description وبنية ملف المعرب Parser Description وبناء مترجم لعملية Scanning فقط.

بنية ملف وصف الماسح :Scanner description file

هو الملف الذي يجب كتابته من قبل المبرمج ويتضمن التوجيهات اللازمة لبناء Scanner:

(1) يتضمن الجزء الأول من البرنامج التعريفات، وتوضع ضمن قوسين %{} للدلالة على أنها تعليمات بلغة الـ C.

```
%{
```

```
#include<stdlib.h>
```

```
%}
```

(2) يتضمن الجزء الثاني التصريح عن القوالب التي سيتم استخدامها للتعرف على تسلسل العبارات المصدرية وإسناد كل عبارة إلى القالب الموافق لها مثلاً:

يمثل القالب التالي قالب الفراغات وله الاسم blank وهو يعني فراغاً واحداً أو أكثر (تشير إشارة + إلى إمكانية أن يكون هناك فراغ واحد أو أكثر).

```
blank [ \t ]+
```

بينما يمثل القالب التالي أحد الأرقام من 1 وحتى 9.

```
number [0-9]
```

أما للتصريح عن قالب يمثل الأعداد المؤلفة من رقم أو أكثر مثل 122 فنكتب:

```
entire {number}+
```

(3) يحتوي الجزء الثالث على القواعد الضابطة لعمل الماسح، والتي ستحدد ماهي المفردة token التي سيعيدها الماسح للمعرب عندما يصادف أحد القوالب في عبارة الدخل المصدرية، للتوضيح:

بفرض أن الماسح صادف مسافة tab أي ثمانية فراغات، وبفرض وجود قالب الفراغات blank الذي تحدثنا عنه، فإن الماسح سيخبر المعرب أنه صادف فراغاً وسيتم تجاهل الفراغ كونه لا يعبر عن أي شيء. نصمم هذه القاعدة كمايلي:

```
{blanc}
```

أما لو صادف الماسح العدد 122، فلا بد له أن يعيد للمعرب المفردة NUM ليدله على أنه صادف عدداً في تسلسل الدخل، لكن يجب أن يعلم المعرب ماهي قيمة هذا العدد ولذلك فإن الماسح يعيد له أيضاً قيمة هذا العدد. يمكن تصميم هذه القاعدة كمايلي:

```
{entire} {
    yylval=atoi(yytext);
    return(NUMBER);
}
```

أما الدالة atoi فهي تأخذ قيمة المتغير yytext الذي يحوي قيمة المفردة المقروءة حالياً من تسلسل الدخل لكنها ستكون من النوع ASCII، لذا تقوم هذه الدالة بتحويل نوع القيمة من النمط string إلى النمط الصحيح integer ليتسنى للمعرب أخذ القيمة الصحيحة، أما عن كيفية ذلك فإن القيمة توضع في المتغير yyval وهذا المتغير هو صلة الوصل بين الماسح والمعرب والذي سيقوم بالتقاط وحفظ قيم المفردات التي يقرأها الماسح من تسلسل الدخل.

(4) القسم الرابع وهو قسم Optional ويمثل التوابيع التي يمكن إضافتها داخل الملف وتضاف حسب الحاجة. مثلاً التابع الرئيسي main يضاف فقط في حالة كنا نريد بناء Scanner بدون Parser.

```
int main()
{
    yylex();
}
```

ملاحظات هامة:

يتم الفصل بين كل قسم من الأقسام السابقة باستخدام الإشارة %.

إذا احتجنا إلى كتابة كود بلغة C، فإنه تتم كتابته بين العلامات %{---}%

يجب تضمين المكتبة stdlib.h ضمن قسم التضمين والتعريف لأننا استخدمنا الدالة atoi منها.

يستخدم المتغير yyval من قبل الماسح ليضع فيه الماسح قيم المفردات التي تتم قراءتها من تسلسل الدخل حتى يتمكن المعرب في ما بعد من استخدامها.

يستخدم الماسح المصفوفة yytext وهي عبارة عن مصفوفة بحجم أعظمي 2000 حرف ليخزن فيها سلاسل المفردات التي يقوم بمسحها من سلسلة الدخل المصدرة.

يمكن لنا تعريف جدول للحروف التي سنستعملها لإنشاء القوالب ومعانيها

.	كل الأحرف باستثناء \n
\n	سطر جديد
*	صفر نسخة أو أكثر من العبارة السابقة لها

+	نسخة واحدة أو أكثر من العبارة السابقة لها
?	نسخة واحدة أو لا شيء من العبارة السابقة لها
^	بداية السطر
\$	نهاية السطر
a b	a أو b
(ab) +	نسخة أو أكثر من السلسلة ab
"a+b"	السلسلة a+b حرفياً
[]	فئة من الأحرف، [a-z] تعني كل الأحرف من a إلى z

وكمثال عما سبق وتطبيقاتها لدينا الجدول التالي:

التطابقات	العبارة
abc	abc
ab,abc,abcc,abccc,...	abc*
abc,abcc,abccc,...	abc+
abc,abcbcb,abcbcbcb,...	a(bc)+
a,abc	a(bc)?
a,b,c	[abc]
أي حرف بين a و z	[a-z]
a,-,z	[a\-,z]
-,a,z	[-az]
حرف أو أكثر (بما في ذلك الأعداد)	[A-Za-z0-9]+
الفراغات	[\t\n]+
أي شيء باستثناء a و b	[^ab]
a,^,b	[a^b]
a, ,b	[a b]
a أو b	a b

لنبدأ بالبرنامج الأول:

سنحتاج في هذا البرنامج فقط لبناء ملف Scanner Description File ولذلك سنقوم بترجمة ملف Scanner وليس Parser وسنقوم بوضع التابع الرئيسي Main في ملف Scanner description File.

تمثل التعليمات الآتية بنية ملف **Scanner Description File**:

```
% {
/* need this for the call to atof() below */
#include <stdlib.h>
% }
```

Stdlib for atof function

```
DIGIT [0-9]
ID [a-z][a-z0-9]*
```

Two templates one for digits and the
another for identifiers

```
%%
```

المتحول ID يبدأ بحرف حصراً يمكن أن يليه حرف أو أكثر أو رقم أو أكثر أو حروف وأرقام أو لا شيء أبداً

```
{DIGIT}+ {
printf( "An integer: %s (%d)\n", yytext,
atoi( yytext ) );
}
```

الاستجابة لقالب الأرقام الصحيحة:
طباعة رسالة تفيد بأن الماسح صادف رقم صحيح ويتم تحويل قيمته من محرف إلى رقم باستخدام الدالة atoi وإعادة قيمته للبنية التي استدعت الماسح (افتراضياً المعرب هو من يستدعي الماسح لكن في مثالنا لا يوجد)

```
{DIGIT}+"."{DIGIT}* {
printf( "A float: %s (%g)\n", yytext,
atof( yytext ) );
}
```

الاستجابة للأرقام ذات الفاصلة

الفاصلة

```
if|then|begin|end|procedure|function|exit|else|for|while {
    printf( "A keyword: %s\n", yytext );
}
```

الاستجابة للكلمات المحجوزة

```
{ID}    printf( "An identifier: %s\n", yytext );
```

الاستجابة لقالب المتحولات ID

```
"+"|"-"|"*"|"|"  printf( "An operator: %s\n", yytext );
```

الاستجابة للعمليات الحسابية

```
"{"[^\n]*"}" /* eat up one-line comments */
```

الاستجابة لتعليق مدخل من قبل المستخدم
يعني أن أي سطر يكتب بين قوسين { } يعتبر تعليق والاستجابة
الطبيعية للتعليقات في البرامج هي (لا شيء)

```
[ \t\n]+ /* eat white spaces */
```

White spaces الفراغات البيضاء
وهي فراغ أو أكثر أو سطر جديد والاستجابة لها هي أيضاً (لا شيء)

```
. {printf( "Unrecognized character: %s\n", yytext );}
```

النقطة تعني إذا صادف Scanner أي رمز أو تركيب ولم يتم مطابقته مع أي من القوالب
السابقة فيتم اعتباره محرف غير معروف ويتم طباعة رسالة تفيد بذلك مع طباعة
المحرف بالاستعانة بالمصفوفة yytext التي تتضمن رموز الدخل المسووحة

```
(([!]*[^\n]*[!]))
```

هنا أيضاً التعليقات لكن تمثل التعليقات التي تمتد على أكثر من
سطر وليس التعليق السطري

```
%%
int yywrap(){return 1;}
int main()
{
    yylex();
}
```

التوابع:

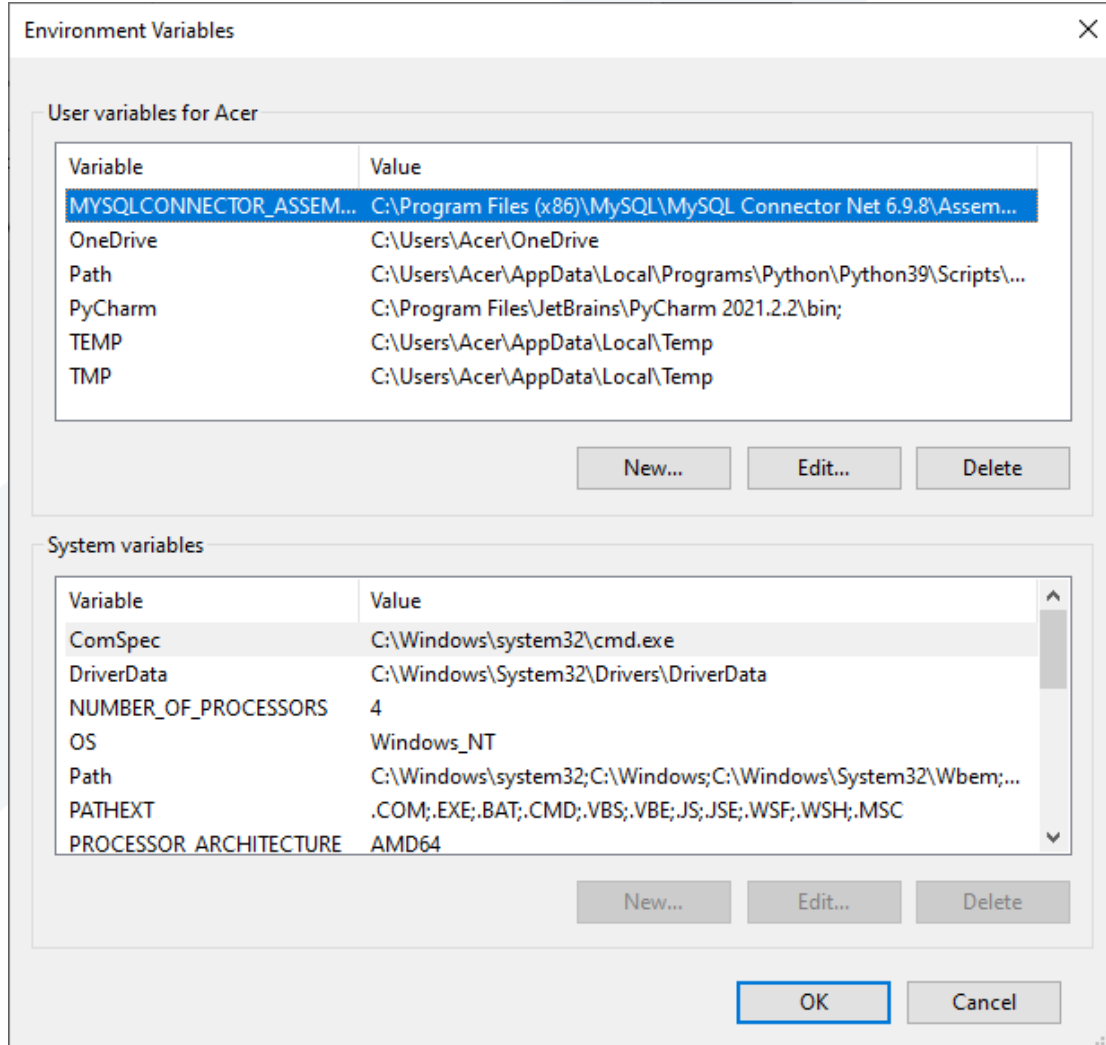
yywrap() وهذا التابع يعيد قيمة 1 عندما ينتهي scanner مسح جميع رموز الدخل
المدخل من قبل المستخدم
التابع الرئيسي main ومنه تنطلق عملية الترجمة وهنا المترجم يتضمن فقط عملية
scanning لذلك نستدعي scanner باستخدام الأمر yylex
أما لو كان المترجم يتضمن مرحلة إعراب كان تم استدعاء parser أي كنا كتبنا yyparse

المراحل العملية لإنتاج المترجم:

المرحلة المسبقة وهي تنفذ مرة واحدة فقط خلال الجزء العملي:

بعد تنصيب Lex, BISON, DevC++ يجب إضافة مسار Bin لهذه البرمجيات إلى متغيرات البيئة Environment Variables حتى نستطيع تنفيذ الأكواد من أي مسار داخل الجهاز.

نفتح ابدأ وفي مربع البحث نكتب Environment Variables أو باللغة العربية متغيرات البيئة (حسب لغة الجهاز) فيظهر خيار Edit Environment Variables نختاره فنحصل على النافذة الآتية:



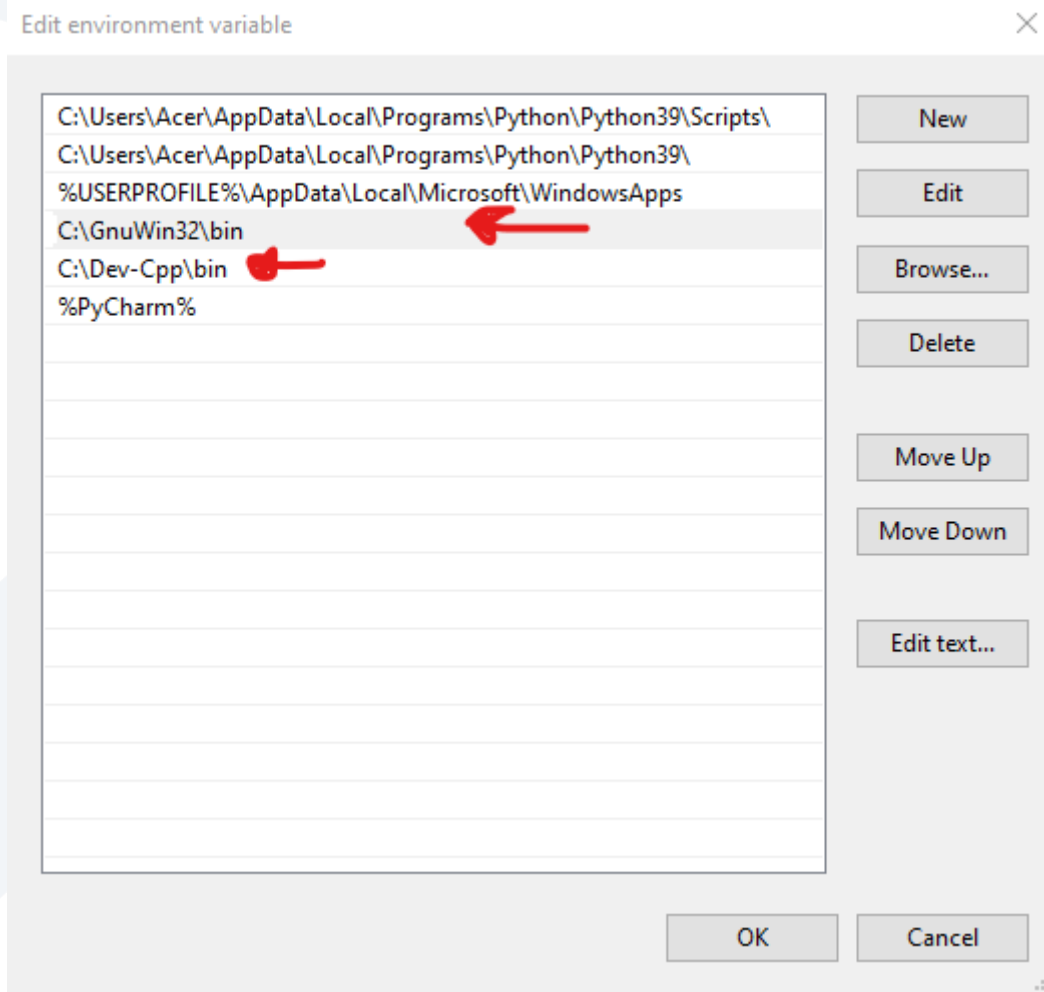
وفي متغيرات المستخدم نذهب للمتغير Path ونقوم بضغط زر edit أو تحرير: ونضغط على زر New ثم نضيف مسار مجلد Bin ل Lex , Bison وهو:

C:\GnuWin32\bin

ثم مسار DevC++ وهو:

C:\Dev-Cpp\bin

ثم موافق



والآن نبدأ بخطوات إنشاء المترجم:

- 1- نكتب هذه التعليمات داخل ملف نصي ونقوم بحفظه في المسار التالي باسم lecture1.L أي نغير امتداد الملف النصي من .txt إلى .L.

C/Lex_Yacc/examples/lecture1



lecture1.l

- 2- ننشئ ضمن نفس المسار ملف نصي باسم lex1 ونكتب في التعليمات الآتية:

flex lecture1.1

pause

نحول امتداد الملف من .txt إلى .bat. بحيث عند النقر المزدوج عليه يتم تنفيذ الملف مباشرة.



lex1.bat

نحصل بنتيجة تنفيذ هذا الملف على ملف lex.yy.c وهو يمثل ملف scanner.



lex.yy.c

3- ننشئ ضمن نفس المسار ملف نصي باسم compile ونكتب في التعليمات الآتية:

```
gcc lex.yy.c -o lecture1.exe
```

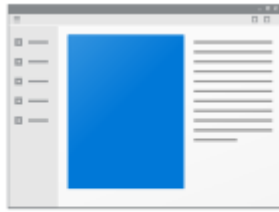
pause

تمثل هذه التعليمة ترجمة ملف lex.yy.c باستخدام مترجم GCC واستخدام بيئة DevC++ لتوليد مفسر بامتداد .exe. يمثل المترجم النهائي المطلوب.



compile.bat

4- بعد تنفيذ الملف compile.bat سنحصل على ملف lecture1.exe ضمن نفس المسار الذي نعمل فيه ويمثل المترجم النهائي وبالنقر المزدوج يبدأ بالعمل.

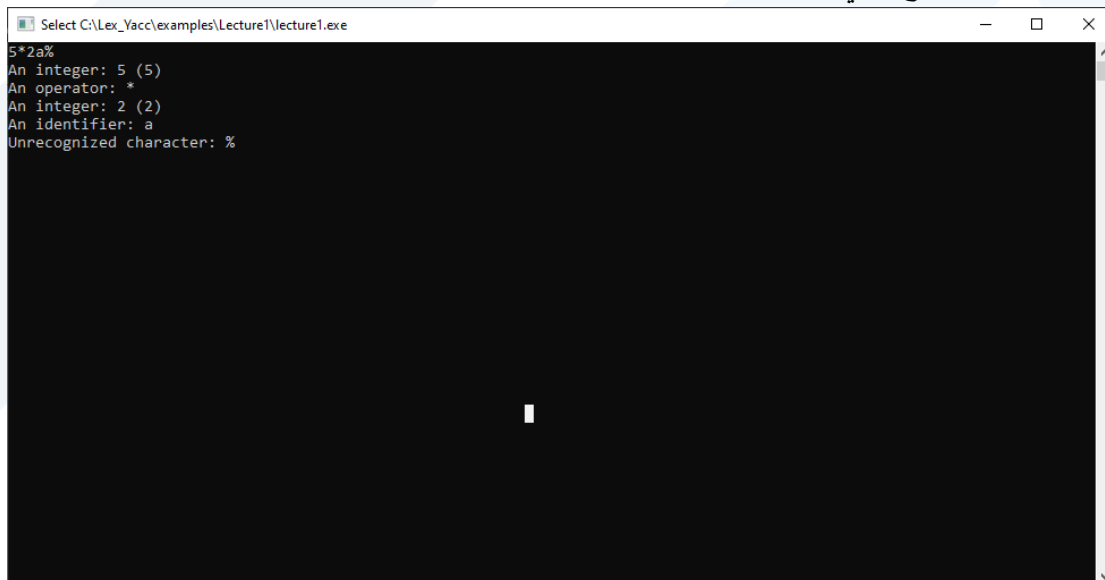


lecture1.exe

5- نختبر المترجم بإدخال العبارة الآتية ثم نضغط Enter

5*2a%

وسنحصل على الخرج الآتي:



```
Select C:\Lex_Yacc\examples\Lecture1\lecture1.exe
5*2a%
An integer: 5 (5)
An operator: *
An integer: 2 (2)
An identifier: a
Unrecognized character: %
```

تعرف الـ Scanner على كل المحارف والأرقام أما الرمز الذي لا يطابق أي قالب لديه فقد قام بطباعة رسالة unrecognized character.

انتهت الجلسة - د. علي ميا ، م. رشا شباني