الجمهورية العربية السورية
هيئة الطاقـة الذريـة

هـ ط ذ س- هـ ن/ ت د ع 704
تشرين الثاني 2006

تقرير عن دراسة علمية حاسوبية
قسم الهندسة النووية

إعداد واجهة تخاطبية لاستخدام الكود CITATION في بيئة النوافذ

إعداد

الدكتورمحمد البرهوم
المهندس نزار زيدان

A Visual Fortran User Interface for CITATION code

M. Albarhoum, N. Zaidan
Atomic Energy Commission of Syria, Nuclear Engineering Department, P. O. Box 6091
Damascus - Syria

Abstract

A User Interface is designed to enable running the CITATION code under Windows. Four sections of CITATION Input File are arranged in the form of 4 interfaces, in which all the parameters of the section can be modified dynamically.
The Help for each parameter (item) can be read from a general Help for the section which, in turn, can be visualized upon secelting the section from the program general menu.

Key Words

Program, Visual FORTRAN, CITATION, Interface, Windows.

المستخلص

يوصف هذا التقرير كيفية إعداد البرنامج CITINT الذي يعمل كواجهة لتشغيل الكود CITATION في بيئة النوافذ. تمت برمجة أربعة مقاطع من ملف دخل هذا الكود وأصبح بإمكـان المسـتخدم تغييـر قـيم متغيرات هذه المقاطع ديناميكياً.يمكن الاطلاع على مدلول متغيرات كل مقطـع مـن المقـاطع الأربعـة المذكورة من خلال خدمة مساعدة تظهر حين طلب تعديل المقطع المطلوب من قبل المستثمر.

الكلمات المفتاحية

برنامج، الفورتران المرئي، الكود سايتيشن، واجهة، بيئة النوافذ.

قائمة المحتويات

العنوان الصفحة

قائمة الأشـكال

العنوان **الصفحة**

قائمة الرموز والمصطلحات والتعاريف

1. **استديوالمطور (Developer Studio):** هوتسمية لبيئة عمل الفورتران المرئي. تمكن هذه البيئة من الدخول لتنفيذ الأوامر عبر حركة الفأرة دون السماح بعرض سطر الأوامر بغية تنفيذ الأوامر منه [1].
2. **سطر الأوامر (Command Prompt):** واجهة خاصة يمكن من خلالها كتابة الأوامر باستعمال لوحة المفاتيح والضغط على الزر ENTER بدلاً من تنفيذها بوساطة حركة الفأرة.
3. **المشروع (Project):** هوالوحدة الأساسية في تنظيم استديوالمطور للبرامج والتطبيقات التي يبنيها المستخدم باستخدام لغة الفورتران المرئي وغيره عموماً وباستخدام المترجم 6.1 خصوصاً. يتكون المشروع من جميع الملفات الأصلية التي يستخدمها البرنامج، إضافة إلى كل المعطيات اللازمة لبناء مشروع. تتكون عملية البناء من تعريف المشروع، ووضع الخيارات له، وكتابة البرنامج أوبناء المكتبة اللازمة له.

4. **تشكيلة (Configuration):** هي الكائن الذي يحدد بنية ملف الخرج النهائي الثنائي (Binary)المبني ضمن مشروع ما، أما بالنسبة لمشروع ما فالتشكيله تقوم ببناء معطياته وتعريفه انطلاقاً من ملفاته الأصلية، حيث تحدد التشكيلة نوع التطبيق المراد إنشاؤه، وأرضية تشغيله (Platform)، وإعدادات الأدوات المستخدمة حين البناء.
5. **مجال العمل (Spacework):** هوإطار تنظيمي لعمل المشاريع ذي درجة أعلى من المشروع بحيث يمكن لمجال العمل أن تضم أكثر من مشروع فيها ولا يحصل العكس.
6. **واجهة (Interface):** إطار الكتروني يمكّن المستخدم (مستخدم البرنامج) من التعامل مع كائنات أخرى عن طريق تغيير معطيات أوإدخال أخرى جديدة في حقول خاصة واضحة المعالم والتعريف في الواجهة.
7. **الكود CITATION:** برنامج حاسوبي معد لحل قضايا نترونية في المفاعل النووي[2].
8. **المقطع (في الكود CITATION):**هومجموعة من المعطيات من طبيعة متماثلة (أوأحياناً مختلفة) موسومة بوجود بطاقة بداية تبدأ بها هذه المعطيات ولا ينتهي المقطع حتى ورود سطر كامل فارغ في ملف دخل هذا الكود أووجود سمة لبداية مقطع جديد. أما بطاقة البداية فهي مؤلفة من ثلاثة أرقام طبيعية حصراً (لا تحتوي على فاصلة عشرية).

1-مقدمة

تشكل الكودات النووية أداة العمل الأساسية للمهندس النووي العامل في أحد مجـالي تصـميم أو تحليـل المفاعلات النووية ، فهي وسيلة التخمين و التقدير و الحساب النهائي في كافة حساباته سواء التصـميمية منها أو التحليلية. و قد يتكرر استخدامها لمرات عديدة قي اليوم الواحد و المسألة الواحدة ، مـا يفـرض استخداماً مريحاً و تعاملاً بسيطاً معها و إلا سئم المستخدم التعامل معها و حاول تجنـب اسـتخدامها ، أو سعى إلى استبدالها بأخرى أكثر سهولة في الاستخدام.

لا بد إذاً من الاتجاه نحو تسهيل استخدام هذه الكودات بشكل عام ، و التركيز علـى تسـهيل اسـتخدام الأساسية منها بشكل خاص. تقوم فكرة التسهيل هذه على إعداد واجهات واضحة و سهلة الاسـتخدام يـتم بموجبها تعامل المستخدم مع الكود من تشغيل و تعديل للمعطيات دون الدخول في تفاصيل ملفات دخل هذا الكود.

و هذه الفكرة ليست جديدة ، إذ أن هذا العمل ليس الأول من نوعه في هذا المضمار ، إلا أن الجديد الذي يستحق الذكر هو أن غالبية الكودات مكتوبة بلغة الفورتران و العمل هذا يركز على إنشاء واجهـة بلغـة الفورتران بدلاً من إنشائها بلغات أخرى. ففي العادة يتم إنشاء الواجهة ، إضافة إلى خدمات التمثيل البياني للمعطيات ، بلغة متخصصة بهذه الأشياء كلغة البيسـيك المرئـي (Visual Basic) [3]أو الـ ـ سـي (C++) أو غيرها ، و قلما يحصل أن تكون لغة برمجة الواجهة هي لغة برمجة الكـود أيضـاً. لا بـل نستطيع القول أنه لم يحصل لنا أن رأينا أحداً صنع الواجهات بلغة الفـورتران المرئـي علـى مسـتوى المؤسسة العلمية التي نعمل بها على الأقل ، على أن المؤسسات الأخرى لا تهتم بهذه اللغة أصلاً.

لقد دأبنا في الماضي على البرمجة بلغة الفورتران لأنها كانت لغة برمجة الغالبية المطلقة مـن الكـودات النووية ، و اليوم نجد أن هذه اللغة القديمة قد تم تطويرها لتستوعب إمكانات اللغات الحديثة ، و بالتـالي وجدنا أن الاستمرار في استخدامها في برمجة الكودات و الواجهات أمر جيد ، كمـا أنـه يحـل مجمـل المشاكل التي تنجم عن اختلاف لغتي برمجة الكود و لغة الواجهة.

لا بد في هذا السياق من ذكر بعض خصائص هذه اللغة:

1- السهولة في الاستخدام ،

2- قدرة اللغة على التعامل مع كودات مكتوبة بلغات أخرى مثل فيجوال باسيك و فيجوال c++ ،

3- قدرة اللغة الفائقة في التعامل مع التوابع الرياضية.

4- إمكانية اللغة من الدخول على وظائف بيئة النوافذ (Windows) و استخدامها جزئياً أو كلياً.

2- لمحة حول المترجم "Visual FORTRAN"

يستخدم الفورتران المرئي بيئةَ التطوير المرئية الخاصة بشركة مايكروسوفت، والتـي تسـتخدمها لغـة C++ المرئية، كما يستخدم وسائل التطوير والبرمجيات الخاصة بالاستديو المرئي (Visual Studio)هذا المترجم مزود بكل ما يحتاجه المبرمج من أدوات لعرض البرنامج وتصحيحه وتنفيـذه. فـيمكن كتابـة برنامج بعدة أشكال، وربطه بعدة أنواع من المكتبات وذلك بالعمل من بيئة التطوير المرئية أومن سـطر الأوامر.

3- كيفية كتابة برنامج (مشروع)

إن استديوالمطور ينظم التطوير والبرامج ضمن ما يسمى بـ " مشاريع" (Projects). يتكون المشـروع من جميع الملفات الأصلية التي يستخدمها البرنامج، إضافة إلى كل المعطيات اللازمـة لبنـاء مشـروع. تتكون عملية البناء من تعريف المشروع، ووضع الخيارات له، وكتابة البرنامج أوبناء المكتبة اللازمة له.
يمكن لكل مشروع أن يعرف تشكيلة أوأكثر(Configuration) تبنى انطلاقاً من ملفاته الأصلية، حيـث تحدد التشكيلة نوع التطبيق المراد إنشاؤه، وأرضية تشغيله (Platform)، وإعدادات الأدوات المستخدمة حين البناء.
تكون المشاريع محتواة فيما يسمى بـ "مجال عمل" (workspace)، لذا يجب تحديد مجال العمـل التـي يستخدم فيها المشروع إن كانت جديدة أم قديمة. يمكن لمجال العمل أن يحتوي على أكثر مـن مشـروع، عندما تفتح مجال العمل فإن بيئة التطوير تعرض شاشة "File View " التي تبين بدورها العلاقات بـين الملفات المكونة للمشروع المعين. يجري عرض الوحدات المتكاملة (Modules) وملفات التضمين (Include files) وكذلك الملفات المكتبية اللازمة بشكل أوتوماتيكي كـ "تابعات" (Dependencies).
يمكن تحديد خيارات البناء من صندوق حوار المشروع (Project menu Settings) لكل مـن كامـل المشروع أوبعض التشكيلات أولبعض الملفات فقط.
عند اكتمال تحديد كل من ملفات المشروع والتشكيلات اللازمة يمكن حينئذٍ بناء المشـروع عـن طريـق الأوامر الموجودة في قائمة Build. يمكن أيضاً تطوير التطبيق من بيئة التطوير المرئية واستخدامه فـي سطر الأوامر بمساعدة ملف Makefile.

4- أنواع المشاريع

في لغة الفورتران المرئي يجب تحديد نوع المشروع عند تشكيله، ويجب تشكيل مشروع لأجل كل ملـف تنفيذي من النوع الثنائي(Binary executable file). إن نوع المشروع معلومة هامـة : فهـي تحـدد بعض الخيارات التي ستتخذها بيئة التطوير المرئية للمشروع ، كذلك الخيارات التي سيستعملها المتـرجم لترجمة الملف الأصل، وكذلك المكتبات الستاتيكية التي سيستخدمها الرابط (Linker) لبناء المشـروع، إضافة إلى أماكن ملفات الخرج.. الخ.

هناك ستة أنواع من المشاريع الممكنة التشكيل والعمل بها باستخدام الفورتران المرئي نوجزها فيما يلي:

4-1 تطبيق فوتران شاشي (FORTRAN Console Application)

يتألف هذا التطبيق من واجهة واحدة دون إمكانية الرسم. لا يحتاج هذا النوع من التطبيقات إلـى خبـرة خاصة بالبرمجة (انظر الشكل1).تكون الملفات التنفيذية من النوع (exe.*).

4-2 تطبيق فورتران معياري بياني (FORTRAN Standarad Graphics Application)

هذا النوع من التطبيقات وحيد النافذة أيضاً، لكنه يوفر إمكانية الرسم (في نظام الـدوس فقـط). تتـراوح صعوبة البرمجة هنا بين الصغر والاعتدال حسب الرسم المطلوب، وطريقة تعامل المسـتخدم المطلوبـة (الشكل2). تكون الملفات التنفيذية من النوع (exe.*).

4-3 تطبيقات وندوز مصغرة (FORTRAN Quick Win Applications)

هي تطبيقات بعدة شاشات مع إمكانية الرسم. تتراوح صعوبة البرمجة هنا، كالتطبيقات الآنفة الذكر، بـين الصغر والاعتدال حسب الرسم المطلوب وطريقة تعامل المستخدم المطلوبة (الشكل2). تكون الملفـات التنفيذية من النوع (exe.*)

4-4 تطبيقات وندوز (FORTRAN Windows Application)

هي تطبيقات بعدة شاشات مع إمكانية الرسم. تتطلب برمجتها خبرة سابقة جيدة، ومعرفة بإجراءات بيئـة النوافذ (Win 32 routines API)، انظر الشكلين3 و 4.

يلاحظ أيضاً في الشكل5 تطبيق آخر من هذا النوع يمكّن المستخدم من الرسم بوساطة الفأرة حيث يمكـن استخدام الزر الأيمن للرسم (رسم الخطوط)، والزر الأيسر لمحوها.تكون الملفات التنفيذية مـن النـوع (exe.*).

5-4 تطبيقات المكتبات الستاتيكية (FORTRAN Static Library)

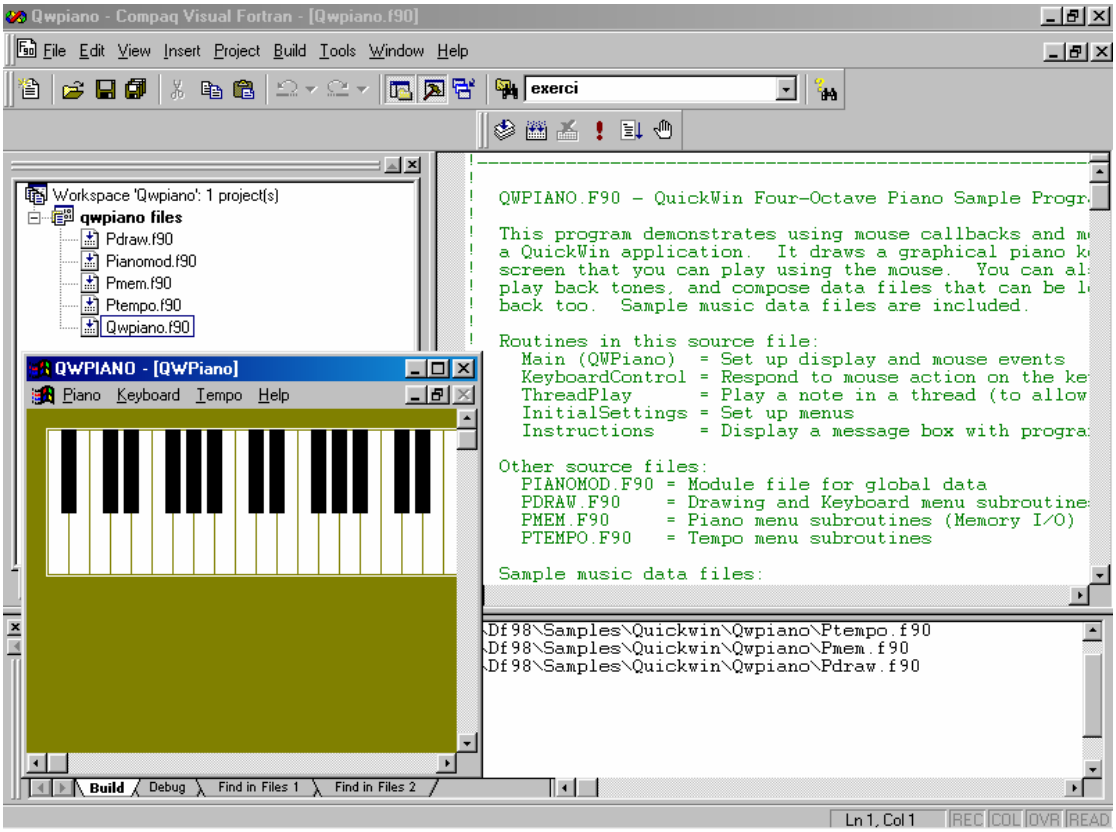
هي تطبيقات مكتبية معدة للربط ضمن ملفات تنفيذية من النوع (*.exe).

6-4 تطبيقات المكتبات الديناميكية (FORTRAN Dynamic Link Library)

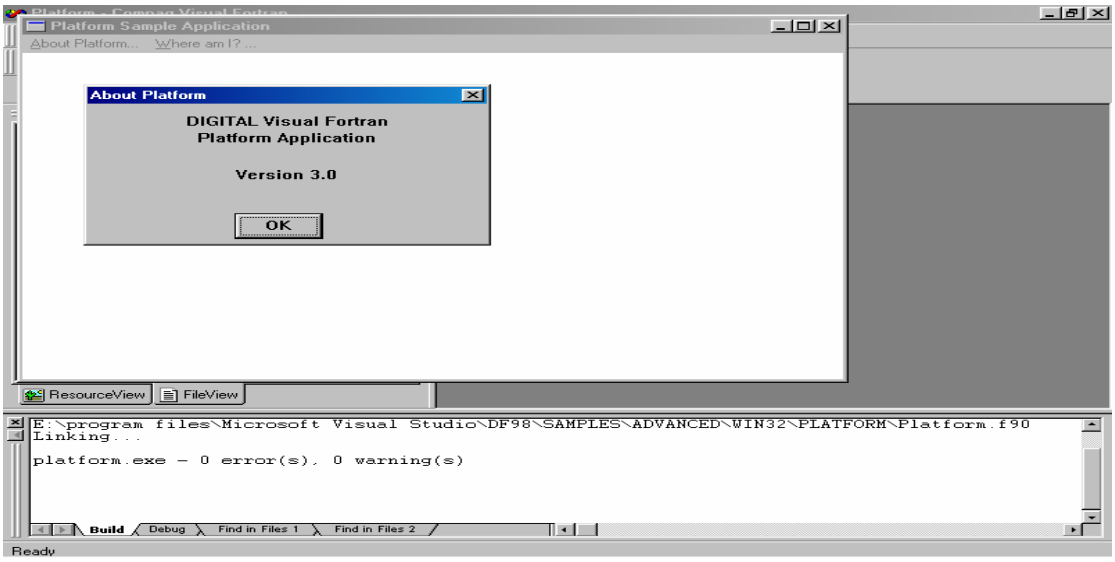
هي تطبيقات مكتبية معدة للربط الديناميكي (أثناء تنفيذ الملف التنفيذي).ملفاتها من النوع (*.dll).

```
Program Celsius Table: Prints simple Fahrenheit-Celsius table!
program celsius_table
implicit none
real Fahrenheit, Celsius
print *,'  Fahrenheit     Celsius'
print *,'-------------------------'
do Fahrenheit = 22, 222, 10
Celsius = (5.0/9.0) * (Fahrenheit-32.0)
print '(F13.0,F12.3)',Fahrenheit,Celsius
end do
end
```

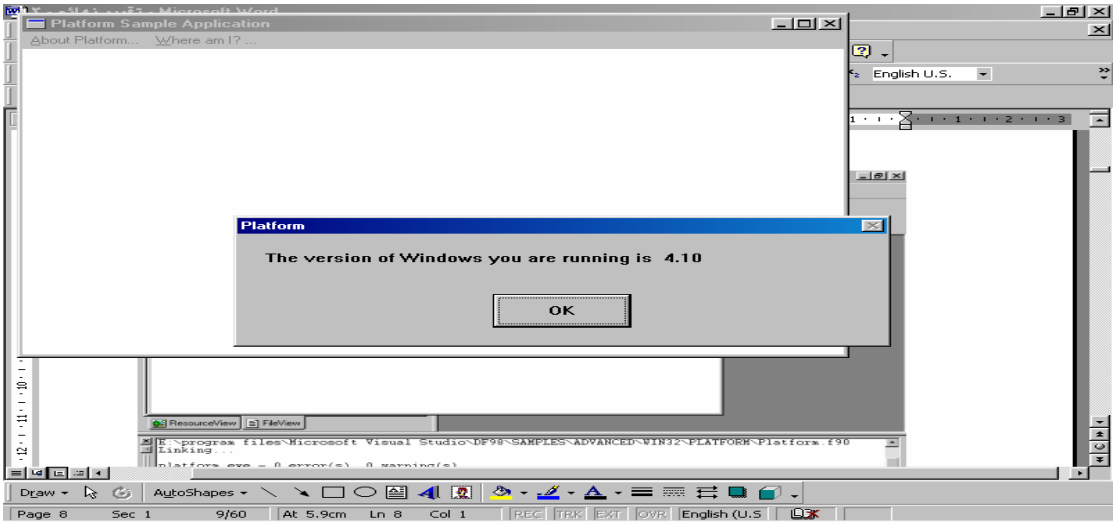
الشكل1 برنامج فورتران مرئي من النوع" شاشي" يطبع جدولاً يقارن بين قيم درجات الحرارة في الميزانين المئوي والفهرنهايت.



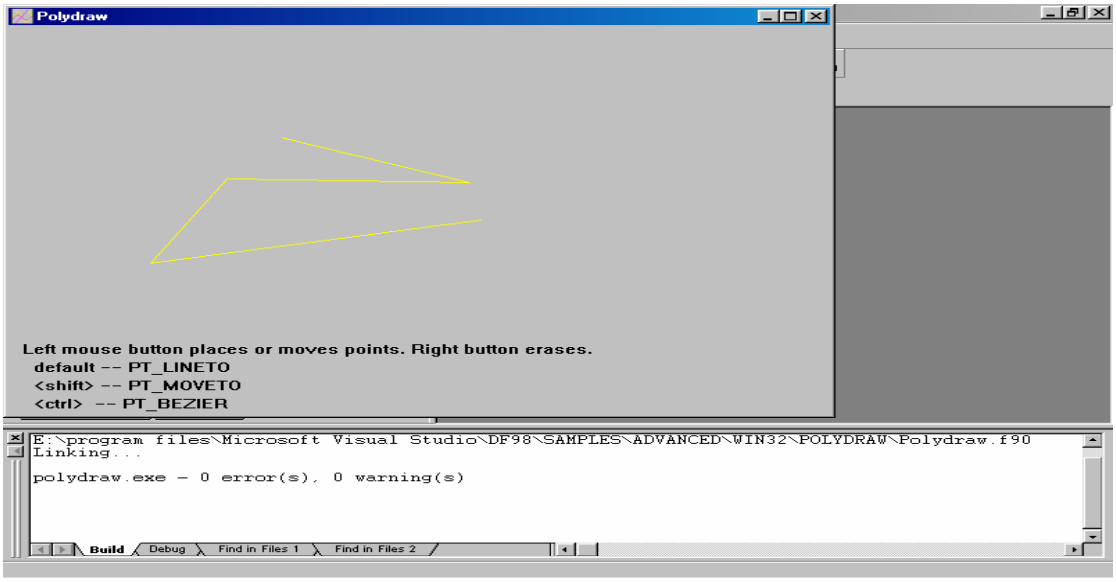
الشكل2 النوع الثالث من المشاريع الممكنة باستخدام بيئة الفورتران المرئي.



الشكل 3 نموذج من تطبيقات الوندوز الممكنة البرمجة بلغة الفورتران المرئي (حالة عرض الرصيف(platform)).



الشكل4 نموذج من تطبيقات الوندوز الممكنة البرمجة بلغة الفورتران المرئي (حالة عرض نسخة نظام النوافذ المستخدم).



الشكل 5 نموذج آخر من تطبيقات الوندوز الممكنة البرمجة بلغة الفورتران المرئي
(حالة برنامج الرسم بواسطة الفأرة).

5- اختيار نوع المشروع للبرنامج

مما تقدم عرضه يبدوواضحاً أن اختيار نوع المشروع سوف يكون واقعاً على النوع الثاني حتـى النـوع الرابع. النوع الثاني جيد من حيث المبدأ، لكنه لا يوفر إمكانية استخدام تطبيقات الوندوز (بيئة النوافـذ)، أي لا يؤمن دخولاً على مكتبة هذه التطبيقات رغم حاجتنا إليها في تشغيل الكود CITATION. وبالتالي فالخيار هذا غير ملائم ويبقى خيارنا محصوراً بين الثالث والرابع من الأنواع المذكورة سابقاً للمشاريع.
الخيار الثالث يؤمن دخولاً جزئياً إلى مكتبات بيئة النوافذ، ولكنه يفي بالغرض بالنسبة لجميع حاجاتنا لهذه البيئة. إذاً يمكننا اختيار أحد النوعين الثالث والرابع لهذه الغاية.

5-1 صفحة الخيارات الرئيسية

في هذه الصفحة تظهر الخيارات الأساسية للبرنامج CITINT والتي نستطيع أن نجملها بما يلي (انظـر الشكل6):

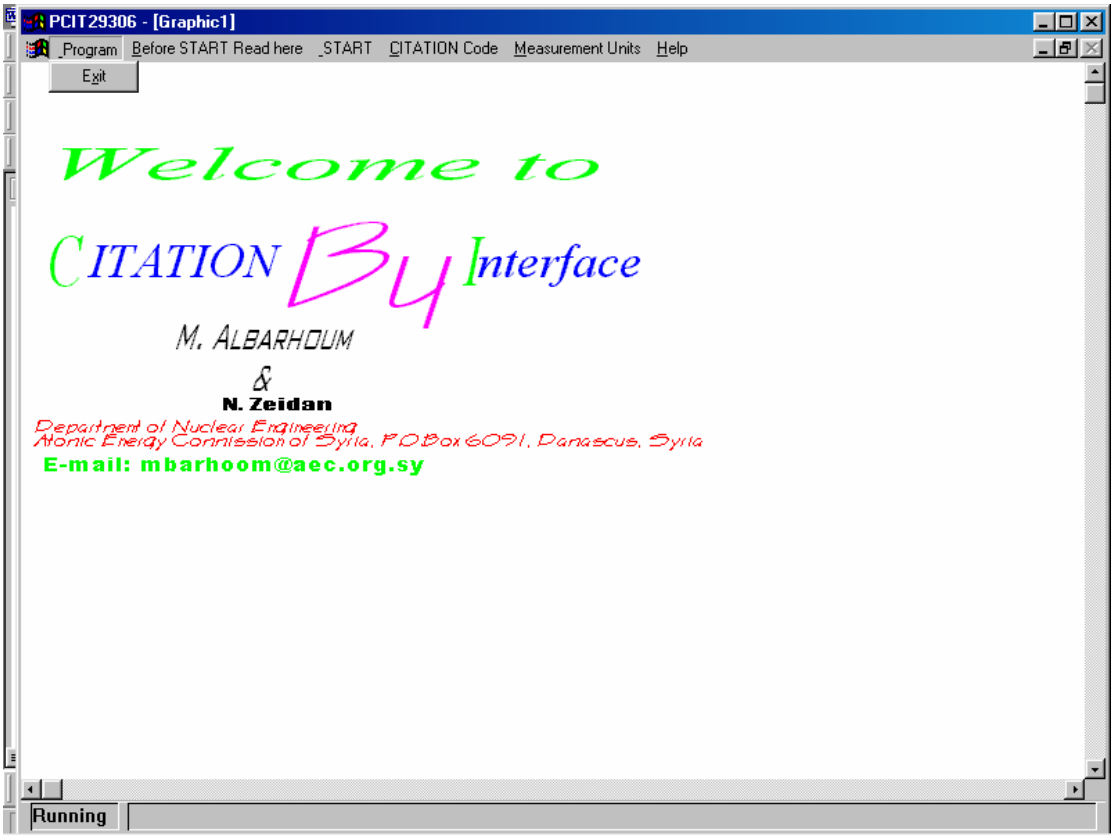
5-1-1 قائمة البرنامج (Program): ويشمل هذا الخيار إمكانية إنهاء البرنامج في أية لحظة يختارهـا المستخدم.

5-1-2 القائمة المنسدلة الثانية " Before START read here": تقوم بتنبيه المستخدم على أنـه يتعين عليه تعريف واحدة القياس من القائمة المنسدلة الخامسة قبل اختبار اسم المقطع المراد تعديله.

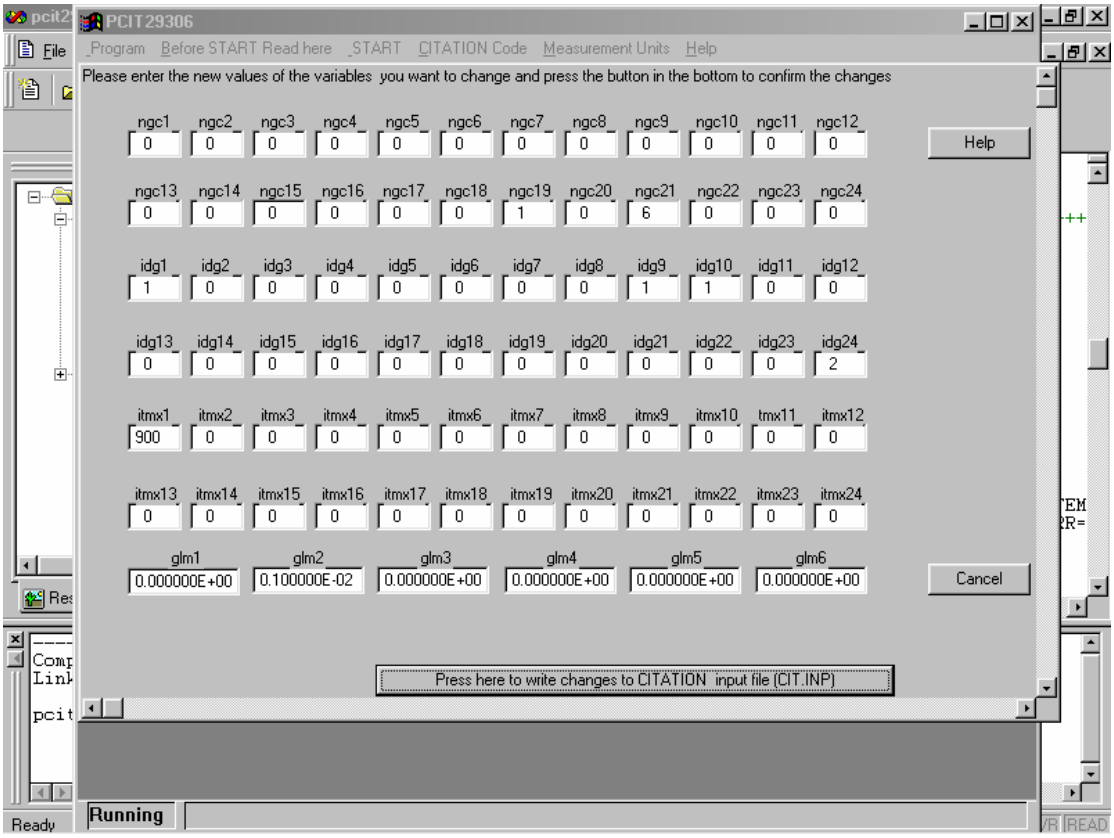
5-1-3 قائمة START: هي القائمة المنسدلة الثالثة و تحتوي على أسماء المقـاطع الممكـن تعـديلها (بعضها غير مفعل آنياً كالمقطع "000").هناك أربعة مقاطع يمكن تعديلها كما يلي:

5-1-3-1 المقطع "000": وهوالآن غير مفعل بسبب عدم استخدامه في المسائل العاديـة المحلولـة باستخدام الكودCITATION.

5-1-3-2 المقطع "001 ": وهومؤلف من أربع بطاقات (انظر الشكل 7).



الشكل6 واجهة البرنامج الرئيسية



الشكل7 واجهة المقطع الأول المؤلف من أربع بطاقات فرعية.

البطاقة الأولى: متغيراتها من النوع طبيعي (NGC) وعددها أربعة وعشرون متغيراً ، و تمثل متغيرات التحكم (خاصة بمقادير خرج البرنامج)[2].

البطاقة الثانية: متغيراتها من النوع طبيعي (IDG) وعددها أربعة وعشرون متغيراً، وتمثل أيضاً خيارات التحرير (Edit Options). تظهر المتغيرات مكتوبة في ناحية من الخانة المخصصة لها على الشاشة غير متوسطة لها ولا منتحية زاويتها اليسرى ولا اليمنى.

البطاقة الثالثة: متغيراتها من النوع طبيعي (ITMX) وعددها أربعة وعشرون متغيراً. وتمثل مجمل عدادات لضبط عدد التكرارات وزمن الآلة وغيره.

البطاقة الرابعة: متغيراتها من النوع حقيقي (GLM) وعددها ستة متغيرات، وتمثل حدوداً مفروضة على قيم معامل التضاعف النتروني.

يجب الانتباه عند تعديل القيم الافتراضية (المأخوذة من ملف دخل الكود CITATION قيد التعديل) إلى كتابة القيمة الجديدة ابتداء من أقصى اليسار المسموح به ضمن الخانة (أومستطيل الإدخال المعني).

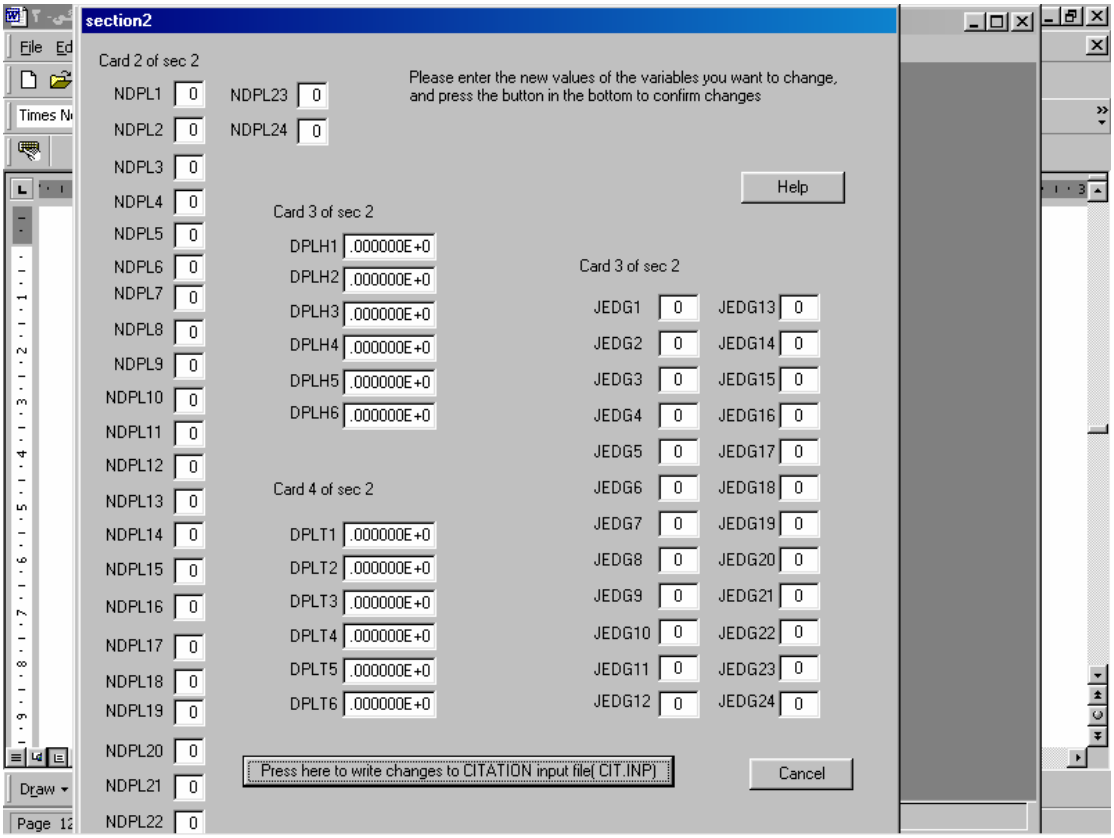
3-3-1-5 المقطع "002": يتألف هذا المقطع من أربع بطاقات موصفة فيما يلي:

البطاقة الأولى: متغيراتها من النوع طبيعي (NDPL) وعددها أربعة وعشرون متغيراً. تمثل بطاقات التحكم في قصة استنفاذ الوقود (Depletion History Control Card).

البطاقة الثانية: متغيراتها من النوع حقيقي (DPLH) وعددها ستة ، و تمثل معطيات تـاريخ الوقـود النووي المستخدم في المفاعل (History Data).

البطاقة الثالثة: متغيراتها من النوع حقيقي (DPLT) وعددها سـتة و تمثـل أزمنـة اسـتنفاذ الوقـود (Depletion Times).

البطاقة الرابعة: متغيراتها من النوع طبيعي (JEDG) وعددها أربعة وعشرون ، وتمثل خيارات تحرير وكتابة الكميات المحسوبة أثناء حساب قصة استنفاذ الوقود في المفاعل (Depletion History).



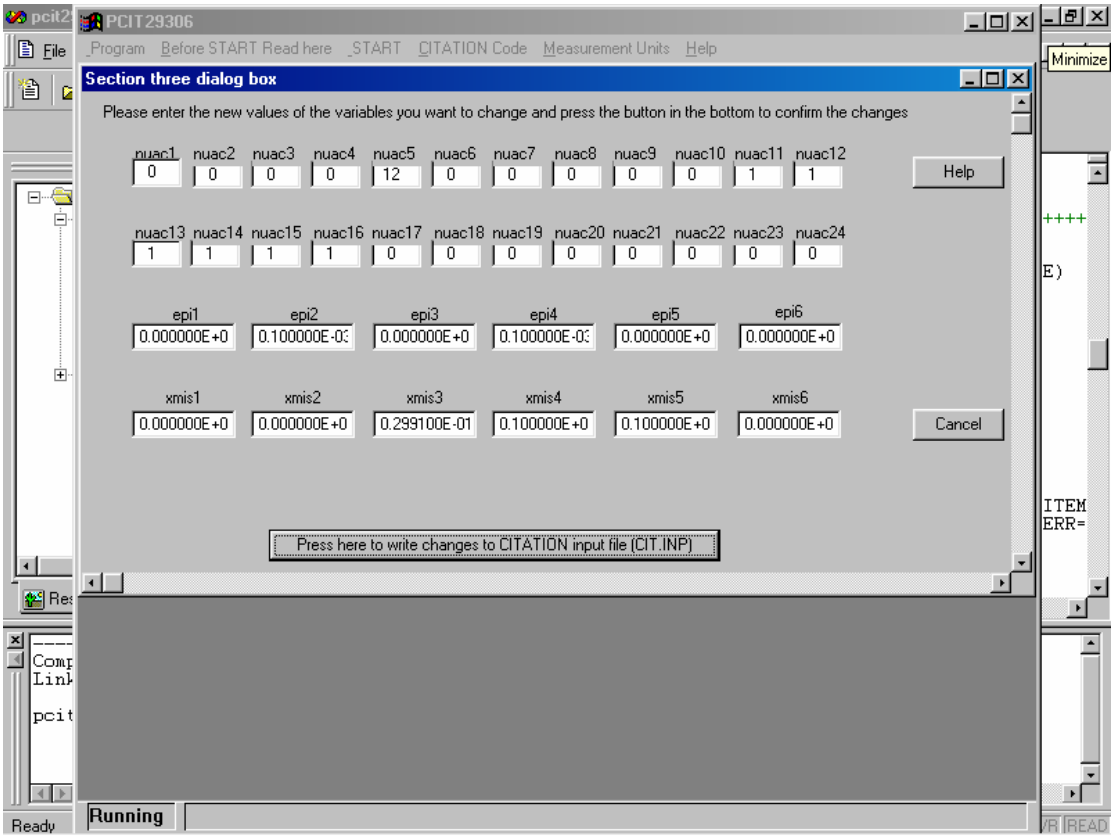
الشكل8 واجهة المقطع الثاني المؤلف من أربعة بطاقات فرعية.

5-1-3-4 المقطع "003": يتألف هذا المقطع من ثلاث بطاقات (انظر الشكل9)موصفة فيما يلي:

البطاقة الأولى: متغيراتها من النوع طبيعي (NUAC) وعددها أربعة وعشرون متغيراً. تقوم معطيـات هذه البطاقة بتوصيف مسألة حساب التدفق النتروني في المفاعل.

البطاقة الثانية: متغيراتها من النوع حقيقي (EPI) وعددها ستة. تمكننا هذه المتغيـرات مـن فـرض خيارات التقارب أثناء حساب مسألة التدفق النتروني في المفاعل. يشمل التقارب كلاً من تقـارب حـل التدفق وتقارب قيمة معامل التضاعف النتروني في المفاعل.

البطاقة الثالثة: متغيراتها من النوع حقيقي (XMIS) وعددها ستة. تهتم هـذه المتغيـرات بمعطيـات مختلفة تبدأ من بعض القيم والشروط الحدية على المسألة المراد حلها إلى تعيين استطاعة المفاعل الحرارية، إلى معامل التحويل (Conversion Ratio)...الخ.



الشكل9 واجهة المقطع الثالث المؤلف من ثلاث بطاقات فرعية.

5-1-4 قائمة CITATION Code

تشمل هذه القائمة الخيارات التالية:

5-1-4-1 خيار Run

يقوم هذا الخيار بتشغيل الكود CITATION الذي يفترض أن يكون ملفه التنفيذي موجوداً، إضافة إلـى مختلف الملفات اللازمه لتشغيله إذا كان الملف التنفيذي قد تمت صياغته باستخدام مترجم مختلـف عـن الفورتران المرئي، في المجلد الذي يحمل اسم مجال العمل (workspace).

يتم عبر هذا الخيار تشغيل الملف التنفيذي وعند الانتهاء يعود التحكم إلى البرنامج الرئيسي. تجدر الاشارة إلى أن ملفات دخل وخرج الكود CITATION محتواة عموماً في مجال العمل الخاصة بالواجهـة مـا يجعل فتح هذه الملفات سهلاً بعد انتهاء الكود CITATION من العمل.

5-1-4-2 خيار Terminate

يقوم هذا الخيار بإنهاء تشغيل الكود CITATION.

5-1-5 قائمة واحدات القياس: (Measurement Units)

تؤمن هذه القائمة (انظر الشكل10) إمكانية استخدام أكثر من نوع من الواحدات للمعطيات النووية (وبخاصة استطاعة المفاعل) إذا لزم الأمر. وهناك خمسة أنواع من الواحدات:

الـميغاواط MW: هي واحدة مطابقة للواحدات المستخدمة للاستطاعة في الكـود (MW)، وبالتـالي يقوم الكود بتركها على ما هي عليه.

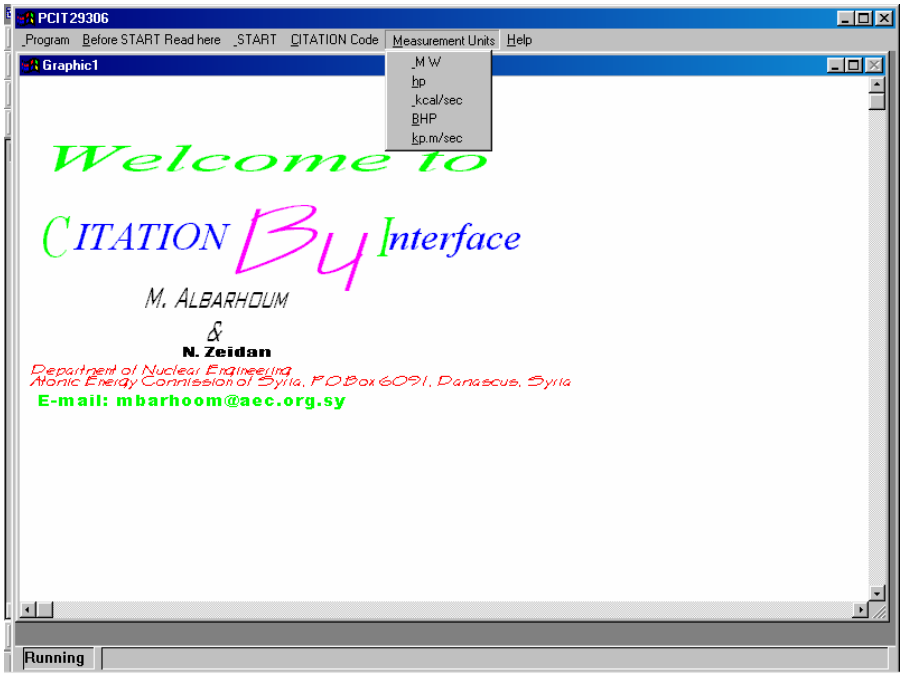
الحصان البخاري (hp)

واحدة الـ كيلوحريرة/ثا (kcal/sec)

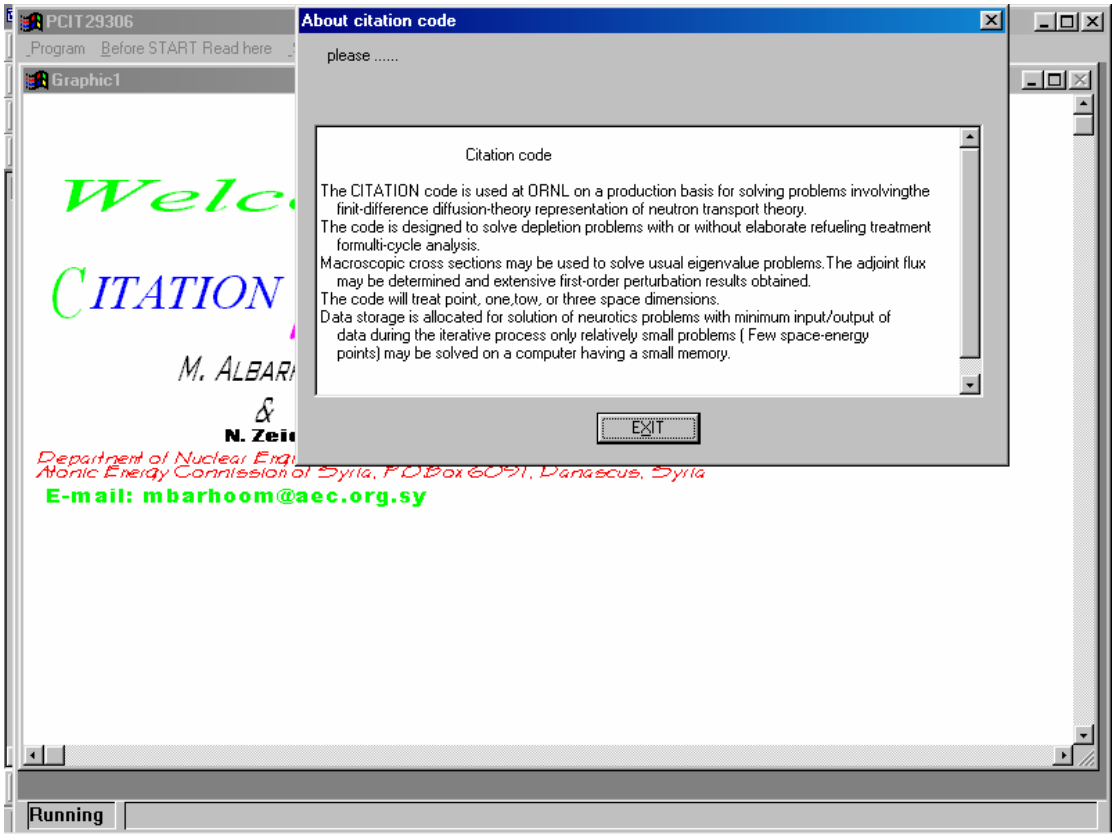
الواحدة الـبريطانية (BHP)

واحدة الكيلوغرام ثقلي*متر/ثا (kp.m/sec): وعند اختيار الواحدة المطلوبة يقوم البرنامج بتحويـل قيم المقادير اللازمة التحويل إلى مكافئاتها في الواحدة المختارة تلقائياً.إذا لم يجر اختيـار أيـة واحـدة فسوف تعتمد واحدات الكود CITATION نفسها (النوع الأول من الواحدات).

5-1-6 قائمة المساعدة (Help): تمكّن هذه القائمة مـن معرفـة بعـض الأشـياء حـول الكـود CITATION، فهي قائمة مساعدة بخصوص الكود بشكل عام (الشكل11).



الشكل 10 قائمة الواحدات الممكن استخدامها في البرنامج (CITINT)



الشكل11 قائمة المساعدة في البرنامج (CITINT).

5-2 خدمة المساعدة في متغيرات المقاطع

لحل مشكلة المساعدة في المقاطع المختلفة فقد جرى إضافة زر في واجهة كل مقطع يمكـن مـن خلالـه إظهار الشروحات والمساعدة اللازمة حول المتغيرات الموجودة في البطاقات المختلفة من المقطع.

5-2-1 متغيرات المقطع "000"

ما زالت هذه المتغيرات غير مفعلة، و بالتالي لا توجد لها مساعدة.

5-2-2 متغيرات المقطع "001"

توجد لها مساعدة للمستخدم (انظر الشكل 12). هذه المساعدة مهيئة على شكل ملف كتبـت فيـه جميـع المعلومات كما هي واردة في دليل الكود CITATION. يمكن تصفح المساعدة إلى الأعلى وإلى الأسفل بحثاً عن المتغير المطلوب (لاحظ أن اسم المتغير مثبت أعلى المتغير في صفحة كل مقطع مـن مقـاطع الكود CITATION).

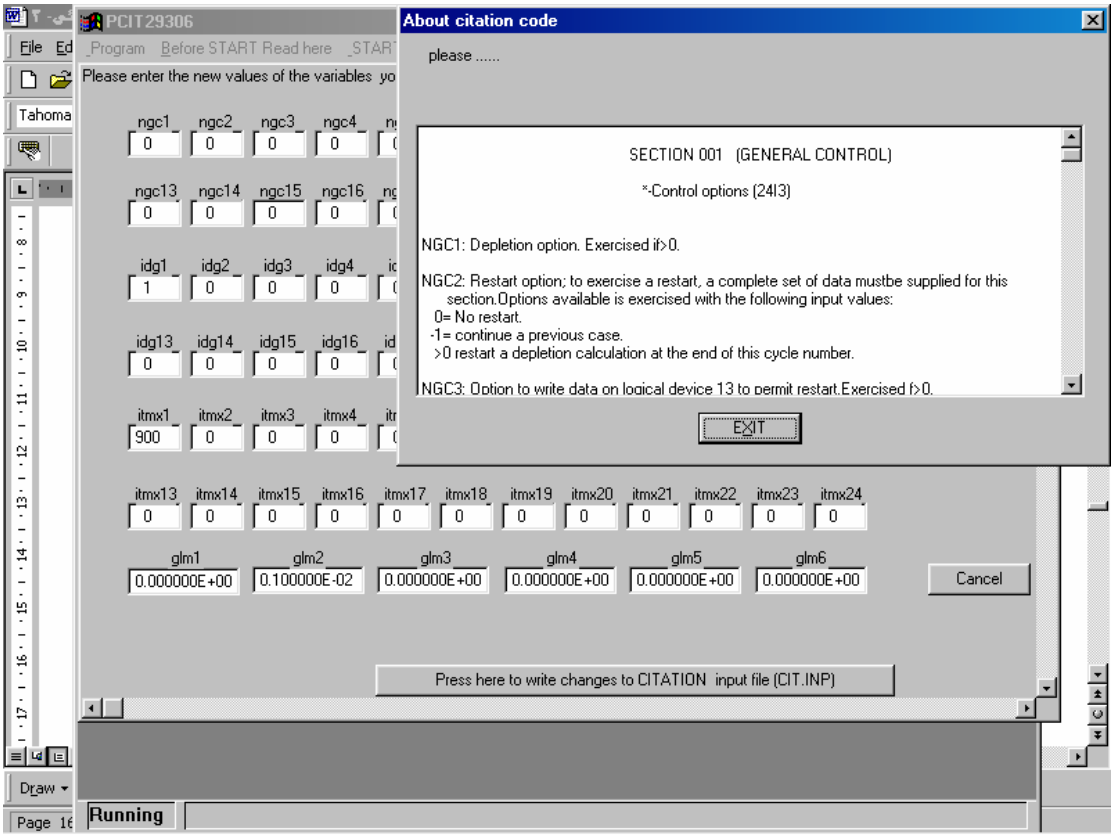
لنلاحظ أيضاً أن الشروحات الموجودة منقولة كما هي في الكود المذكور دون تعديل أوإضافة مـن قبلنـا للأمانة وحتى يترك المجال للمستخدم في فهم معانيها، دون أن نلزمه بكيفية فهمنا لها (إن حصل هنـاك أي اختلاف).

5-2-3 متغيرات المقطع "002"

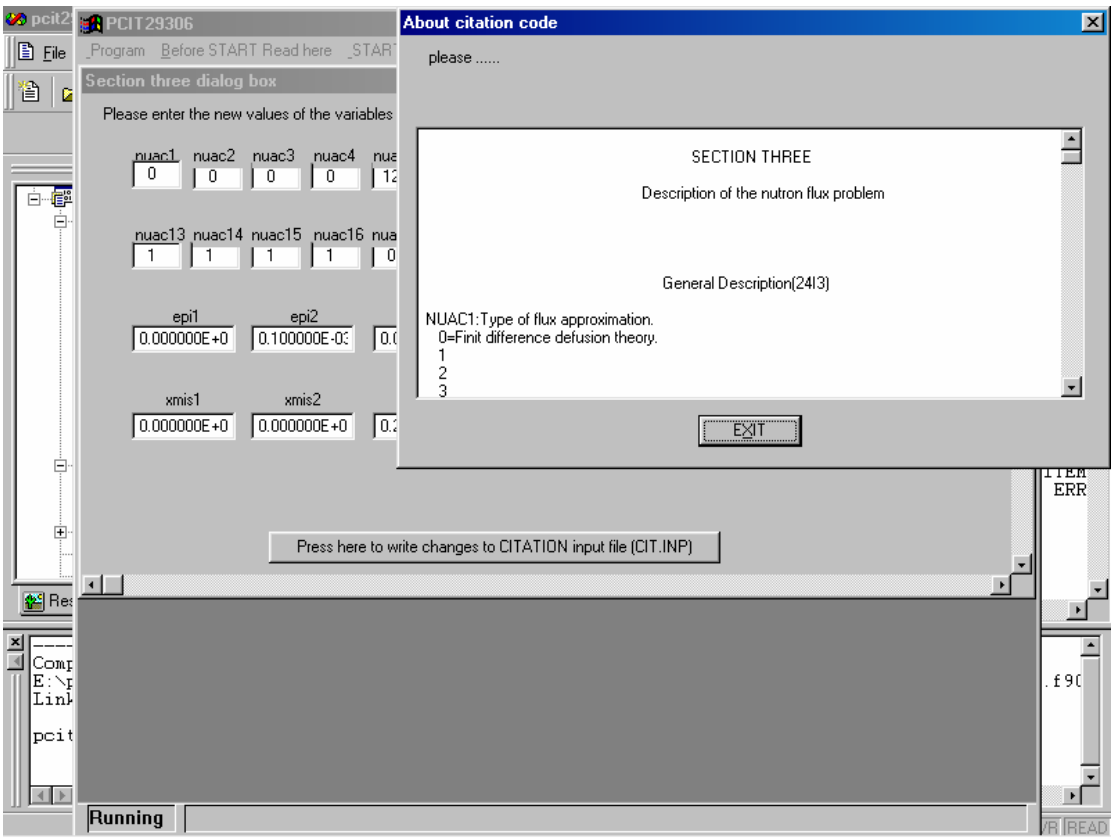
يمكن إعادة ما قيل بخصوص مساعدة المقطع "001" في حال المقطع "002".

5-2-4 متغيرات المقطع "003"

مساعدة المقطع "003" مبنية بالطريقة نفسها التي بنيت بها مساعدة المقطع "001" (انظر الشكل 13).



الشكل12 مظهر مساعدة المقطع "001" عند الطلب من قبل المستخدم.



الشكل13 مظهر مساعدة المقطع "003" عند الطلب من قبل المستخدم.

6- الإجراءات والوظائف التي يتألف منها البرنامج CITINT

6-1 الملف الأول (Mainp.f90)

يقوم هذا البرنامج بتعيين القوائم المنسدلة الأساسية المطلوبة من البرنامج (انظر الفقرة السـابقة)،إضـافة إلى أنه يحدد شكل الواجهة الأساسية الأولى لعمل البرنامج من حيث تحديد أبعاد النافذة الأولـى ومكـان توضعها على الشاشة، وكذلك يحدد شكل صفحة الخيارات الرئيسة. يستدعي هذا البرنـامج عـدداً مـن الإجراءات والوظائف هي:

- **البرنامج الرئيسي** الذي يشغل البرنامج CITINT بشكل إجمالي، وهوبسيط (انظر الشكل14)

```
Program CITINF
CALL WARNADV(10,10,'CITATION interface)
do while (.TRUE.)
end do
END
```

الشكل14 نص البرنامج الرئيسي في البرنامج CITINT

- **برنامج القيم الابتدائية:**البرنامج الذي يعطي القيم الابتدائية للإطـارات ويحـدد الواجهـات بالشـكل المطلوب(الشكل ب-1 في الملحق ب).

6-2 الملف الثاني Sectionone.f90

يقوم هذا البرنامج بتحرير قيم المتغيرات الموجودة في المقطع الأول ووضعها على الشاشة. كما ويمكنـه كتابة القيم الجديدة لهذه المتغيرات في ملف الدخل للكود CITATION عن طريق الضغط علـى الـزر الأسفل في واجهة المقطع "001" (انظر الشكل7). يحوي هذا البرنامج الفرعي عـدداً مـن الإجـراءات والوظائف منها:

- **الإجراء Applyok1:**

ويقوم هذا الإجراء بقراءة القيم الجديدة لمتغيرات المقطع "001" بعد تعديلها من قبل المستخدم.

- **الإجراء Sectiononehelp:**

يقوم هذا الإجراء بعرض ملف المساعدة للمقطع "001" عند طلب عرض واجهة هـذا المقطـع علـى الشاشة من قبل المستخدم.

- **الإجراء Runningcitation**

يقوم هذا الإجراء بتشغيل الكود CITATION عند طلب تشغيله من قبل المستخدم. يشغل هذا الإجراء الملف التنفيذي للكود فقط الذي تمت ترجمته باستخدام مترجم ما ودون إعادة ترجمته باستخدام متـرجم الفورتران المرئي.

- **الإجراء Units**

يقوم هذا الإجراء بالتعديلات المناسبة على متغيرات مختلف المقاطع عندما يحدد المستخدم واحدة قياس مختلفة عن الافتراضية والتي هي واحدات الكود CITATION نفسه.

3-6 الملف الثالث Sectionthree.f90:

يقوم هذا البرنامج بتحرير قيم المتغيرات الموجودة في المقطع الثالث ووضعها على الشاشة. كمـا يمكنـه كتابة القيم الجديدة لهذه المتغيرات في ملف الدخل للكود CITATION عن طريق الضغط علـى الـزر الأسفل في واجهة المقطع "003" (الشكل3). يحوي هذا البرنامج الفرعي عدداً من الإجراءات والوظائف منها:

- **الإجراء Applyok3:**

ويقوم هذا الإجراء بقراءة القيم الجديدة لمتغيرات المقطع "003" بعد تعديلها من قبل المستخدم.

- **الإجراء Sectionthreehelp:**

يقوم هذا الإجراء بعرض ملف المساعدة للمقطع "003" عند عرض واجهة هذا المقطع على الشاشة من قبل المستخدم.

- **الإجراء DIST**

يثبت هذا الإجراء القيم الجديدة المدخلة في مختلف المقاطع ("001"، "003"، "002" و"000") في ملـف دخل الكود CITATION وتقوم مختلف الإجراءات باستدعاء هذا الإجراء حين الحاجة.

4-6 الملف الرابع Sectionzero.f90:

يقوم هذا البرنامج بتحرير قيم المتغيرات والبارامترات الموجودة فـي المقطـع "000" ووضـعها علـى الشاشة. كما يمكنه كتابة القيم الجديدة لهذه المتغيرات في ملف الدخل للكود CITATION عـن طريـق الضغط على الزر الأسفل في واجهة المقطع "003" (الشكل3). يحوي هذا البرنامج الفرعي عـدداً مـن الإجراءات والوظائف منها:

- **الإجراء Applyok0:**

ويقوم هذا الإجراء بقراءة القيم الجديدة لمتغيرات المقطع "000" بعد تعديلها من قبل المستخدم، ولكـن المقطع غير مفعل في هذه النسخة من البرنامج CITINT.

- **الإجراء Sectionzerohelp:**

يقوم هذا الإجراء بعرض ملف المساعدة للمقطع "000" عند عرض واجهة هذا المقطع على الشاشة.

6-5 الملف الخامس sectiontwo.f90:

يقوم هذا البرنامج بتحرير قيم المتغيرات والبارامترات الموجودة فـي المقطـع "002" ووضـعها علـى الشاشة. كما يمكنه كتابة القيم الجديدة لهذه المتغيرات في ملف الدخل للكود CITATION عـن طريـق الضغط على الزر الأسفل في واجهة المقطع "002" (انظر الشكل8). يحوي هذا البرنامج الفرعي عـدداً من الإجراءات والوظائف منها:

- **الإجراء Applyok2:**

ويقوم هذا الإجراء بقراءة القيم الجديدة لمتغيراتت المقطع "002" بعد تعديلها من قبل المستخدم.

- **الإجراء Sectiontwohelp:**

يقوم هذا الإجراء بعرض ملف المساعدة للمقطع "002" عند عرض واجهة هذا المقطع على الشاشة.

6-6 الملف السادس unit.f90:

يقوم هذا البرنامج بتصحيح المتغيرات المحتواة في المقاطع المختلفة عن طريـق اسـتدعاء عـدد مـن الإجراءات المعددة فيما يلي:

- **الإجراء Unitmw:**

ويقوم هذا الإجراء بتفعيل واحدة الاستطاعة الميغاوات من أجل جميع مقاطع الكود CITATION التي يعالجها البرنانمج CITINT.

- **الإجراء Unitcv:**

ويقوم هذا الإجراء بتفعيل واحدة الاستطاعة الحصـان البخـاري مـن أجـل جميـع مقـاطع الكـود CITATION التي يعالجها البرنامج CITINT.

- **الإجراء Unitkcalds:**

ويقوم هذا الإجراء بتفعيل واحدة الاستطاعة الكيلوحريرة /ثا من أجل جميع مقاطع الكود CITATION التي يعالجها البرنانمج CITINT.

- **الإجراء Unitphb:**

ويقوم هذا الإجراء بتفعيل الواحدة البريطانية للاستطاعة من أجل جميع مقـاطع الكـود CITATION التي يعالجها البرنانمج CITINT.

- **الإجراء Unitkpmds:**

ويقوم هذا الإجراء بتفعيل واحدة الاستطاعة الكيلوغرام ثقلي.م/ثا مـن أجـل جميـع مقـاطع الكـود CITATION التي يعالجها البرنانمج CITINT.

7-6 الملف السابع Warnadv.f90:

يحتوي هذا البرنامج الفرعي على الإجراء warnadv الذي يهيئ الواجهة الأساسية للبرنامج (شكل1).

الملف الثامن Aboutprog:

يقوم هذا الإجراء بتفعيل واجهة خاصة تعرف بمنتج البرنامج. وهذه الواجهة مكملة للواجهة التي تعـرض عند تشغيل البرنامج.

الملف التاسع Citationhelp.f90:

يقوم هذا البرنامج بحل مسألة المساعدة في البرنامج CITINT بالنسبة لجميـع الواجهـات (واجهـات المقاطع) عن طريق استدعاء الإجراءات المناسبة لكل مقطع.

7-الملفات المصدر الملحقة بالبرنامج

1-الملف Dialogs.rc

يحتوي هذا الملف على جميع واجهات المستخدم (المسماة واجهات حوار "Dialogs" كواجهات المقاطع عموماً)، وصناديق الحوار أيضاً.

2-الملف Icon1.ico

وهوملف ملحق بالبرنامج يساعد على تحديد الأيقونة المرافقة أثناء عمل الإطار النشط للبرنامج. وفـرت لغة البرمجة فورتران6.1 هذه الميزة بفضل برنامج صغير يساعد علـى تصـميم الأيقونـات، وهوأحـد المصادر الأساسية في فيجوال فورتران6.1.

3-الملف Resource.fd

وهوملف نص يعرف كل متغيرات النوافذ المستخدمة في البرنامج، ويعمل على تأمين وصول البرنـامج إلى النوافذ المطلوبة دون خطأ. يمكّن هذا الملف المبرمج من التواصل مع النوافذ المصـممة مـن قبلـه ويمكّنه أيضاً من السيطرة على النوافذ من حيث- الإزالة والتحرير-لذلك هوملف ملحق أساسي.

8-الملفات الخارجية المستخدمة من قبل البرنامج

1-الملف Cit.in

يمثل هذا الملف الخارجي ملف دخل الكود CITATION الرسمي بالنسبة للبرنامج، وبالتـالي فـالتغيير على المعطيات يتم ويسجل في هذا الملف.

2-الملف Cit.out

يمثل هذا الملف الخارجي ملف خرج الكود CITATION الرسمي بالنسبة للبرنامج، وهذا يحدده الكـود CITATION نفسه وليس للبرنامج علاقة مباشرة به.

3-الملف Sr1.txt

ويستخدم لوضع المعطيات بشكل مؤقت بعد عرضها للتغيير أوالتعديل من قبل المستخدم، وتنسخ منه هذه المعطيات بشكل نهائي في الملف Cit.inp.

4-الملف Sectionzero.txt

يحتوي هذا الملف على نص المساعدة للمقطع "000". يقوم البرنامج بقراءة هذا الـنص وعرضـه علـى الشاشة حين طلب تعديل هذا المقطع من قبل المستخدم.

5-الملف Sectionone.txt

يحتوي هذا الملف على نص المساعدة للمقطع "001". يقوم البرنامج بقراءة هذا الـنص وعرضـه علـى الشاشة حين طلب تعديل هذا المقطع من قبل المستخدم.

6-الملف Sectiontwo.txt

يحتوي هذا الملف على نص المساعدة للمقطع "002". يقوم البرنامج بقراءة هذا الـنص وعرضـه علـى الشاشة حين طلب تعديل هذا المقطع من قبل المستخدم.

7-الملف Sectionthree.txt

يحتوي هذا الملف على نص المساعدة للمقطع "003". يقوم البرنامج بقراءة هذا الـنص وعرضـه علـى الشاشة حين طلب تعديل هذا المقطع من قبل المستخدم.

9- النتائج

نستطيع القول أنه بنتيجة هذا العمل تم التوصل إلى برنامج مكتوب بلغة الفورتران المرئي ويعمل ضـمن بيئة النوافذ، يستطيع أن يكون صلة الوصل بين المستخدم والكود CITATION من حيث إمكانية تعـديل أية قيمة في المقاطع القابلة للتعديل من قبل المستخدم مباشرة وهي محصورة في المقاطع ("000"، "001"، "002"، "003") بالنسبة لجميع المسائل التي لا تتطلب استعمالاً خاصاً للكود CITATION، ولا تتطلـب استخدام هذا الكود في إدارة الوقود (انظر الأشكال 7، 8 و 9) بالإضافة إلى الشكل 6 الذي يمثل واجهـة البرنامج الرئيسة). أما المقاطع التي لا يمكن للمستخدم تعديلها مباشرة، و التي تكون قيم المعطيات فيهـا تابعة للمسألة المراد حلها أوهي قيم مقاطع عرضية، أوتعبر عن طريقة نمذجة المفاعل فإن هذا البرنامج لا يتدخل بها لاختلاف طبيعته وأهدافه عما ذكر (المقطع "008"، المقطع "999"...الـخ –انظـر الفقـرة اللاحقة).

يوفر البرنامج إمكانية إجراء الحسابات في الكود CITATION على أساس واحدة قيـاس الاسـتطاعة المختارة من قبل المستخدم، ولئن كان استخدام هذه الواحدة قليلاً في المقاطع التي يتعامل معها هذا الكـود الآن، إلا أنه يمكن – في حال توسيع هذا البرنامج ليشمل المسائل المستثناة في البند السابق- أن يكون لها استعمال أوسع وفائدة أكبر في مقاطع مثل ("091" و"093")، وغيرها من المقاطع التي يمكن لمعطياتها أن تكتب بأكثر من واحدة قياس.

يوفر البرنامج إمكانية تشغيل الكود CITATION من بيئة النوافذ حتى ولولم يكـن مترجمـاً بوسـاطة مترجم الفورتران المرئي (أي دون العودة إلى موجه دوس – أوما يسمى بـ "سطر الأوامـر") (انظـر الشكلين 15 و16). وهنا يجب التنويه إلى أن مسـألة اسـتخدام الفـورتران المرئـي لترجمـة الكـود CITATION وتشغيله في بيئة النوافذ مباشرة و دون الحاجة إلى متممات تشغيله اللازمة في حال كانت ترجمته باستخدام مترجم آخر غير الفورتران المرئي، هي مسألة من نوع آخر لا تقع ضمن مهـام هـذا العمل، لأن مهمة هذا العمل تنحصر في تشغيل الكود ضمن بيئة النوافذ بغـض النظـر عـن المتـرجم المستعمل لترجمته وبالتالي فلوازمه متأتية من طبيعة المترجم المستخدم في ترجمته ولزومها محصور في تشغيل الملف التنفيذي للكود CITATION.

```
599999999999999999999999999999999999999999999999992
1    32-bit Power for Lahey Computer Systems     1
1   Phar Lap's 386|DOS-Extender(tm) Version 4.1L   1
Inc.   1 ,1  Copyright (C) 1986-92 Phar Lap Software
1          Available Memory = 128368 Kb          1
499999999999999999999999999999999999999999999999993
```

الشكل15 واجهة تشغيل الكود CITATION في بيئة النوافذ باستخدام المترجم Laheyلترجمته.

```
Setting environment for using Visual Fortran tools
>E:\program files\Microsoft Visual Studio\Common\MSDEV98\My Projects
```

الشكل(16) سطر الأوامر بالنسبة لمترجم الفورتران المرئي

10- التوصيات

مما سبق يمكن الاستنتاج أن هذا البرنامج كاف بصيغته الحالية ليكون واجهـة اسـتخدام جيـدة للكـود CITATION بالطريقة التي يستخدم فيها الكود الآن، أما إذا اتسع استخدام هذا الكـود ليطـال إجـراء حسابات إدارة الوقود فسيكون ضرورياً حينها توسيع هذه الواجهة لتشمل مقاطع أخرى كما أسلفنا الـذكر سابقاً (فقرة النتائج) وبالتالي نوصي بأن يتم التقدم باستمارة عمل آخر في حال حصل هذا الاتساع فـي استخدام الكود CITATION .

قد يتعلق هذا الأمر بمدى استخدام كودات أخرى في العمل مثل الكود MCNP، والحاجة إلـى مقارنـة حسابات هذين الكودين في هذه المسألة أوتلك، ولكن هذا الأمر مازال مجهولاً بالنسبة لنا، ونحن على أتـم استعداد لتوسيع مهام هذا البرنامج حين الحاجة.

11- المراجع

[1] Compaq Visual FORTRAN- Professional Edition 6.1.0, Digital Equipment Corporation. Copyright ©1997-1999.
[2] Fowler T.B, Vondy D.R, and Cunningham G.W. (1971) Nuclear Reactor Core Analysis Code: CITATION. ORNL-TM-2496, Rev. 2, July.
[3] معجم الكمبيوتر -الشركة المتحدة للبرمجيات و الحاسب الآلي- القاهرة –مصر(2003).

[4]A. Heydon, M. W. Maimone, J. D. Tygar, J. M. Wing, and A. Moormann Zaremski, "Miró tools," in Proc. 1989 IEEE Workshop Visual Languages , Oct. 1989, pp. 86-91.
[5] Eric J. Golin and Steven P. Reiss. *The specification of visual language syntax*. In Proc. 1989 IEEE Workshop on Visual Languages, pages 105--110, Rome, Italy, 1989.
[6] Margaret Burnett, Dept. of Computer Science, Oregon State University, email: burnett@cs.orst.edu , and David W. McIntyre, Morgan Stanley, email: mcintyre@is.morgan.com

12-الملحق أ

12-1 البرمجة المرئية

نستعرض فيما يأتي بعض التعريفات والكتابات حول البرمجة المرئية لإعطاء القارىء الأرضية العلميـة اللازمة لتفهم خياراتنا البرمجية في هذا العمل.

التعريف الأول:

"1-A visual language manipulates visual information or supports visual interaction, or allows programming with visual expressions. The latter is taken to be the definition of a visual programming language. [Golin90b] [4] "

لنلاحظ أنه حسب هذا التعريف يمكن أن تكون البرمجة المرئية واحدة من ثلاثة اشـياء: إمـا أن تعـالج معلومات مرئية، أوأن تسمح بالتعامل والتفاعل المرئي، أوأن تسمح بالبرمجة باسـتخدام أشـياء مرئيـة. فالتعريف إذاً ليس واحداً، كما يظهر من الكتابة السابقة.

التعريف الثاني:

ولننظر ماذا يقدم لنا هذا الموقع من تعاريف أخرى لهذه اللغة.

"What is a Visual Programming Language?A few representative answers:
2-Visual Programming (VP) refers to any system that allows the user to specify a program in two-(or more)-dimensionsional fashion. [4] conventional textual languages are not considered two dimensional since the compilers or interpreters process them as long, one-dimensional streams. [4] "

لنلاحظ أنه طبقاً لهذا التعريف فإن اللغات المعروفة (الجيل الثالث منها- فورتران- بيسـك،...الـخ) لا تعتبر لغات برمجة مرئية أساساً لأن المترجم يعامل هذه اللغات على أنها صف واحد طويل من المعطيات والمعلومات.

التعريف الثالث:

"3-A visual Language manipulates visual information or supports visual interaction, or allows programming with visual expressions. The latter is taken to be the definition of a visual programming language. Visual programming languages may be further classified according to the type and extent of visual expression used, into icon-based languages, form-based languages and diagram languages. Visual programming environments provide graphical or iconic elements which can be manipulated by the user in an interactive way according to some specific spatial grammar for program construction. [4] "

هذا التعريف هونفسه الملحوظ في (1) في مقدمته، ولكنه أكثر تفصيلاً، إذ يقسم البرمجة المرئيـة حسـب النوع وحسب كمية "التعابير المرئية المستخدمة".

يمكننا الآن تتبع وجهة نظر "Burnet " التالية:

"Visually transformed languages are inherently non-visual languages but have superimposed visual representations. Naturally visual languages have an inherent visual expression for which there is no obvious textual equivalent. [5] "

لنلاحظ في هذه الرؤية نفي كون اللغات المصيرة مرئية، نفي كونها لغات مرئية في الأساس، وأن اللغات التي هي بطبيعتها مرئية ليس لها مكافئ مرئي معروف.

التعريف الرابع:

"4-Visual programming is commonly defined as the use of visual expressions (such as graphics, drawings, animation or icons) in the process of programming. These visual expressions may be used in programming environments as graphical interfaces for textual programming languages; they may be used to form the syntax of new visual programming languages leading to new paradigms such as programming by demonstration; or they may be used in graphical presentations of the behavior or structure of a program. [6] "

في هذه المقولة لـ بيرنت ومك انتاير (McIntyre&Burnett) يظهر جلياً أنه يمكن للغة برمجة نصية (كمعظم لغات البرمجة المعروفة من الجيل الثالث) أن تسمى لغة برمجة مرئية فيما لواسـتخدمت هـذه اللغة تعابير مرئيةَ كواجهات بيانية للبرمجة النصية.

التعريف الخامس:

"5-A visual language is a set of spatial arrangements of text-graphic symbols with a semantic interpretation that is used in carrying out communication actions in the world. "

أما هذا التعريف فهو،كما يبدو، مغاير قليلاً أوكثيراً للتعاريف السابقة، ولكن سنجد تبريراً لهـذا التعريـف عن طريق المناقشة التالية:

"Do we need the word "programming" in that phrase?

Perhaps not. People like to point out languages such as Miro [4] and GIL [5] which are visual specification languages as reasons for saying visual language instead of visual programming language. I think of Miro as a language for programming specifications, so I like the word.
We'll try to avoid using the word "programming" when we don't mean to exclude non-programming visual languages. "

لنلاحظ الآن التعليقات والآراء على السؤال السابق:

"Any comments?
[Fred Lakin says:] Sure. The short answer is, it reminds us of all the other visual languages there are, which should be looked at and learned from. Keeping the word "programming" in the phrase keeps the computer folk from becoming visually provincial, which I see as a real danger.
The longer answer is, people have invented and used many visual languages in the course of history. A fraction of those have anything to do with computers, and an even smaller number represent programs, and an even smaller number of those represent programs and can be executed on a computer. Let's say people have been using visual languages for 10,000 years; and using them for communication about computers for 50, and using them for communication with computers for 30. So you can see how small a percentage of the total numbers of visual languages we are talking about. "

نلاحظ أن حديث اللغات المرئية حديث شائك وقابل للمناقشة كثيراً بحيث لا يمكننا الجزم واعتماد أفكـار معينة، بل يهمنا من مجمل الحديث أن نقول: أننا نستعمل اللغة (لغة الفورتران المرئي لأن الناشر نشرها بهذا الإسم، لا لأننا نريد أن نبرهن على أنها لغة مرئية، فلسنا نحن من قام بتسميتها هكذا، كما أن الحديث

عن اللغة المرئية حديث غير متفق عليه عموماً، حتى أن التسمية بحد ذاتها موضع خـلاف أيضـاً بـين المختصين كما سنبين فيما يأتي.

12-2 حول تسمية البرمجة المرئية

"Is there a better phrase (than VPL) that we could use?
I prefer the term "executable graphics" instead of visual programming languages. Visual programming language is a misnomer. It either means a programming language which we can see, which is trivial, or a language used for programming the behavior of visual things, which is limiting. Executable Graphics expresses a different orientation toward the problem domain: graphics which can be executed. [5]
Paul Lyons says:
I've coined the term "Hyperprogramming" which I think better summarises the capabilities and support provided by visual Programming Languages. We argue for VPLs on practical as well as theoretical basis. The theoretical arguments relate to the greater expressivity and intuitiveness of diagrammatic representations of complex relationships. The practical arguments relate to the availability of sufficient computing power to support the capture and processing of visually expressed diagrams. Specifically, we utilise:
processor speed, to let us do it in real time
high-res graphics, to represent complex diagrammatic notations
mouse input, to create complex diagrammatic notations and
window-based displays, to partition the resulting diagrams into a manageable size.
It's this last point that's the important one. Partitioning big programs to make them more manageable is great, but creates navigational difficulties. These sort of navigational problems have been addressed, for "ordinary" documents, by hypertext systems. Now, "ordinary" hypertext documents are tedious to create because adding all the hyperlinks takes a long time, but there's no such problem with programs, because it's easy for the entry support system to generate the hyperlinks automatically, on-the-fly. As well as providing programmers with simple and consistent navigation techniques, the hyperlinks can be used to automatically update shared information between views.
So I think that VPLs, if they aren't already, will achieve partitioing based on multiple windows, with hyperlinks between the windows connecting shared items of information. Calling them Hyperprogramming languages will reflect this situation, and might reduce the subtle suggestion (inherent in the name VISUAL programming languages) that these languages should eschew text entirely. [5] "

كما هومبين فيما سبق من النصوص فإن هناك اقتراحات لكلمات أخرى أكثر تعبيراً عن المعنى (معنـى البرمجة المرئية)، وتطابقاً في المعنى، من الكلمة المستخدمة حالياً.

12-3 حول أهمية البرمجة المرئية

"3-Doesn't everyone agree that VL is great?
Heck, no! In fact, some pretty well-respected people have nothing but contempt for the visual representation of software. In a very famous article [Brooks87] Fred Brooks says this:
A favorite subject for PhD dissertations in software engineering is graphical, or visual, programming - the application of computer graphics to software design.... Nothing even

convincing, much less exciting, has yet emerged from such efforts. I am persuaded that nothing will.
Of course, Brooks' arguments contain several weaknesses:
He focuses on flowchart-based control-flow diagrams.
He is worried about screen size in pixels. Phil Cox has presented a strong argument why this may not be meaningful.
I think he misunderstands the power of multiple views - not superimposed views.
Another anti-vl quote:
...beware the claims of visual programming. Drawing lines between objects becomes bafflingly web-like. Purely visual programming is not yet and may never be viable. [OBrien93] "

12-4 حول لغتي البيزك المرئي والـC++

"Visual Basic and the entire Microsoft Visual (tm) family are not, despite their names, visual programming languages. They are textual languages which use a graphical gui builder to make programming decent interfaces easier on the programmer.

لا يعتبر هذا الرأي هاتين اللغتين لغتي برمجة مرئية على الرغم من تسميتهما به.

How do you talk about Visual Programming Languages in an ASCII medium (i.e., USENET)?
Good question. Debate over this one continues. Some people on comp.lang.visual suggest that it can be done, citing film criticism as a textual medium talking about a decidedly non-textual medium. Others say that, sure, you can criticize a released film, but how can you talk about a film no one has seen (and by extension, a VPL no one has made programs with)?
Brook Conner (dbc@cs.brown.edu) tends to go with the first team (that meaningful discussion can take place). Textual criticism will not replace actual experience. However, it can still be valuable. After all, text is fundamentally a form of communication, just like movies, animation, hypermedia, and that old standby, speech. The fact that there are some things text does not do well is probably why many of us are interested in VPLs in the first place, but I don't think anyone on this group would say "Text is useless". "

12-5 ما هومفهوم واجهة المستخدم البيانية (Graphical User Interface)؟

يعرف معجم الكمبيوتر [3]هذا الاصطلاح كما يلي:

"واجهة المستخدمين البينية التخطيطية: هي وسيلة تستخدم في أنظمة التشغيل الحديثة، مثـل النوافـذ 95، والتي تتيح الاتصال بالحاسوب من خلال الأيقونات، icons، وقوائم السـحب والإسـقاط drag and drop menu أي إمكانية التعامل مع الحاسوب من خلال الصور. " ، أمـا موسـوعة ENCARTA البريطانية فتعرفه على الشكل:

"Graphical User Interface or GUI, in computer science, a type of display format that enables the user to choose commands, start programs, and see lists of files and other options by pointing to pictorial representations (icons) and lists of menu items on the screen. Choices can generally be activated either with the keyboard or with a mouse. "

13-الملحق ب

13-1-مشاريع البرمجة من النوع: Quick Win Applications

يؤمن هذا النوع من المشاريع مرونة أكبر مما هي عليه مرونة المشاريع من النوع الثاني (راجع الفقرة)، ذلك لأنه يؤمن فتح نوافذ متعددة (MDI) والانتقال فيما بينها أثناء عمل البرنامج،كما يمكن تحديد مواقع وأبعاد هذه النوافذ على الشاشة.

تسمح مكتبة Quick Win باستخدام عدد من الإجراءات البرمجية (Routines) التي تؤمن بناء تطبيقات تستخدم نسخة مصغرة عن واجهة Windows مع الفورتران المرئي (بينما يؤمن النوع الرابع من هذه المشاريع استخدام واجهة Windows كاملة –API- مع الفورتران المرئي مع إمكانية الاستدعاء المباشر لهذه الإجراءات دون استخدام Quick Win لبناء البرنامج).

يكون لتطبيقات Quick Win واجهات متعددة وبالتالي يمكن استخدام قوائم منسدلة في رأس الواجهة من الأعلى، وقوائم الحالة في الجزء الأدنى من الواجهة.

يؤمن استخدام Quick Win مجموعة من القوائم، ولكل قائمة مجموعة من الموضوعات يمكن التحكم بها عن طريق واجهات كويك ون (Quick Win APIs). يمكن للتطبيقات التي تستخدم Quick Win أن تتعامل مع صناديق الحوار عن طريق الدخول إلى DFLOGM.F90 ويمكن لتطبيقات WINDOWS أن تستدعيها مباشرة.

تقوم بيئة التطوير المرئية بشكل أوتوماتيكي بتضمين مكتبة QuickWin فيها عند تحديد نوع المشروع كتطبيق لـ QuickWin لتسمح باستخدام الإمكانيات البيانية لهذا النوع من المشاريع.

يمكن بناء هذا النوع من المشاريع من سطر الأوامر بخيارات معينة للترجمة (/libs:qwin).تجدر الإشارة إلى أنه لا يمكن تحويل تطبيق من النوع Quick Win إلى آخر من النوع DLL.

13-2-حول بعض الإجراءات المتاحة في مكتبة Quick Win

لدخول البرنامج المكتوب بلغة الفورتران المرئي إلى مكتبة Quick Win لا بد للمستثمر من إيراد عبارة USE DFLIB في البرنامج.

INITIALSETTINGS function.

يمكن لنظام ويندوز (أولـ كويك وين) أن يستدعي هذه الوظيفة لتعيين نافذة وقوائم فيها، وتحديد أبعاد وموقع هذه النافذة قبل رسم هذه النافذة.

WINEXIT

يستطيع نظام وندوز أن ينهي برنامجاً ما عن طريق استخدام هذه الوظيفة.

INSERTMENUQQ function

لإنشاء قائمة في الإطار الفعال الابتدائي للبرنامج يمكن استخدام هذه الوظيفة، وقد كانت تستخدم الوظيفـة قبلهاAPPENDMENUQQ.

13-3-البرنامج الابتدائي المشغل للواجهة CITINT

نورد فيما يلي استخدام الوظيفة 1 من الفقرة السابقة في حال البرنامج CITINT.

```
LOGICAL(4) FUNCTION INITIALSETTINGS( )
USE DFLIB
USE DFLOGM
!USE MT
INCLUDE 'resource.fd'
LOGICAL(4) result
TYPE (qwinfo) wincitation
!external  aboutprog,sectionone,sectionthree,help_citation,WARNADV,sectiontwo
external  aboutprog,sectionone,sectionthree,help_citation,WARNADV, SECTIONTWO& ,
SECTIONZERO,  UNITMW,  UNITCV,UNITKPMDS,UNITKCALDS,
UNITBHP,RUNNINGCITATION
COMMON /NGHH/ NUN

NUN=1
  wincitation%x = 50
  wincitation%y = 0
  wincitation%w = 700
  wincitation%h = 600
wincitation%type = QWIN$SET
  i = SetwsizeQQ( QWIN$FRAMEWINDOW, wincitation )
  result = INSERTMENUQQ(1,0, $MENUENABLED, '& Program'C, NUL )
  esult = INSERTMENUQQ(1,1, $MENUENABLED, 'E&xit'C, WINEXI )
  result = INSERTMENUQQ(2,0, $MENUENABLED, '&Before START Read here'C, NUL )
  result = INSERTMENUQQ(2,1, $MENUENABLED, '&Select the Measurement Units&
before you choose the section to be modified'C,NUL )
    result = INSERTMENUQQ(3,0, $MENUENABLED, '& START'C, NUL  )
    result = INSERTMENUQQ(3,1, $MENUENABLED, 'Section 0'C, sectionzero  )
    result = INSERTMENUQQ(3,2, $MENUENABLED, 'Section 1'C, sectionone    )
    result = INSERTMENUQQ(3,3, $MENUENABLED, 'Section 2'C, sectiontwo )
    result = INSERTMENUQQ(3,4, $MENUENABLED, 'Section 3'C, sectionthree )
    result = INSERTMENUQQ(4,0, $MENUENABLED, '&CITATION Code'C, NUL  )
    result = INSERTMENUQQ(4,1, $MENUENABLED, '&Run'C,RUNNINGCITATION )
    result = INSERTMENUQQ(4,2, $MENUENABLED, 'Terminate'C, WINEXIT )
    result = INSERTMENUQQ(5,0, $MENUENABLED, '&Measurement Units'C, NUL )
    result = INSERTMENUQQ(5,1, $MENUENABLED, '& M W'C,UNITMW )
    result = INSERTMENUQQ(5,2, $MENUENABLED, '&hp 'C,UNITCV )
    result = INSERTMENUQQ(5,3, $MENUENABLED, '& kcal/sec'C, UNITKCALDS )
    result = INSERTMENUQQ(5,4, $MENUENABLED, '&BHP 'C, UNITBHP )
    result = INSERTMENUQQ(5,5, $MENUENABLED, '&kp.m/sec 'C, UNITKPMDS )
    result = INSERTMENUQQ(6,0, $MENUENABLED, '&Help'C, NUL )
    result =INSERTMENUQQ(6,1,$MENUENABLED,'&LEARN CITATION
        CODE'C,HELP_CITATION )
```

```
result = INSERTMENUQQ(6,2, $MENUENABLED, 'About'C, ABOUTPROG )
INITIALSETTINGS= result
END FUNCTION INITIALSETTINGS
```

الشكل ب- 1 نص البرنامج المحدد للقيم الابتدائية للواجهات والشاشات في البرنامج CITINT

Syrian Arab Republic
Atomic Energy Commission(AECS)
Damascus P. O. Box 6091

Report on Scientific Informatic Study
Department of Nuclear Engineering

A Visual Fortran User Interface for CITATION code

Dr . M. Albarhoum
Eng . N. Zaidan

AECS - NE \ RSS 704 November 2006