بسم الله الرحمن الرحيم

السلام عليكم و رحمة الله و بركاته

اليوم باذن الله سوف نتابع الجزء الثاني من المعلومات المهمة التي يجب على كل مبرمج ان يلم بها

نبدأ على بركة الله

1 - اللغات المترجمة و المفسرة

كثير منا عند رغبته في الدخول الى عالم البرمجة تواجهه هذه النقطة في بداية طريقه و العجيب ان الكثيرين من المبرمجين ربما يفهم هذه النقطة جيدا بعد فترة من تعمقه في عالم البرمجة عموما

اللغات المترجمة (Compilation Language) :

هي تلك اللغات التي تعتمد على ترجمة البرنامج مباشرة الى كود بلغة الآلة لا يفهمه الا الحاسب .

اللغات المفسرة (Interpreted Language) :

هي اللغات التي تتم عملية ترجمتها الى كود بلغة الآلة في كل مرة نحاول فيها استخدام البرنامج .

طيب ربما التعريفات القصيرة التي في الأعلى بسطت الموضوع جدا كل ما في الامر لغة تترجم مرة واحدة في حين لغة تترجم كل مرة نستعمل فيها البرنامج من هنا نستنتج ان البرامج المترجمة اسرع من البرامج المفسرة لان الكمبيوتر لا يطر في الاولى الى النظر الى كود البرنامج كل ما عليه هو التنفيذ اما في الاخرى فالبرنامج تتم ترجمته ثم يبدا في تنفيذه لذا فالبرامج الكبيرة عادة لا تكتب بلغات مفسرة و اظن السبب واضح ☺ في حين ان اكتشاف الاخطاء في البرامج المفسرة اسهل بكثير من اكتشافها في البرامج المترجمة فالكود تتم مراجعته في كل مرة يتم فيها تشغيل البرنامج

من اشهر الامثلة على اللغات المترجمة C++

و من اشهر الامثلة على اللغات المفسرة Java

و بالحديث عن Java فهي لغة تعتمد على اسلوب جميل يسمح لها بالعمل على جميع المنصات بسهولة فالجافا تعتمد على ما يسمى java vertical machine هذه الاداة السحرية تقوم بتشغيل البرامج المكتوبة ب java على مختلف الانظمة و الفكرة هنا سهلة (في الفهم)
كل ما يحدث ان الاداة تحول كودك الى كود وسيط (byte code) بدوره تحوله الاداة الى

كود الآلة و تعتبر ايضا لغة مترجمة في نفس الوقت عند تنفذ البرنامج يعني استفادة من التفسير في اكتشاف الاخطاء و من الترجمة في سرعة الاداء .

وفي رأيي الشخصي الجافا هي الخيار الافضل لبرامج الموبايل
لكنها ليست الافضل لبرامج الحواسيب
و طبعا هذه وجهة نظر و بالتاكيد تختلف من مبرمج لآخر

لا ادري لكن عندي اعتقاد ان كل مبرمجين الجافا حذفوا هذا الكتاب بعد هذه الجملة
يا ترى الاحسن احذفها و اريح نفسي ؟؟؟ ☺

2 - ادوات تطوير البرامج

سنستعرض في هذا الجزء ما نحتاجه لنبدأ في أي لغة

1 - المترجم أو المفسر : و هو البرنامج الذي سبق و تحدثنا عنه و لكل لغة مترجمها او مفسرها الخاص و لا بد من معرفة كيفية استخدامه و كل هذا يكون عن طريق موقع اللغة .

2- محرر النصوص : هناك العديد من هذه المحررات في الانترنت حمل احدها و ابدا بكتابة برامجك عليها لكن ما فائدتهاالمحرر احد اهم مهامه هي مساعدتك في التمييز بين الكلمات المحجوزة للغة و بين الاكواد التي تكتبها ناهيك عن ترتيب الكود بعض هذه المحررات يكون فيها ادوات تسهل عليك استدعاء المكتبات او الدوال اشهر و ابسط هذه المحررات Notepad الموجود في ويندوز .

3- برامج التنقيح : من اسمها عبارة عن برامج تساعدك في اختبار برامجك و اكتشاف الاخطاء و هي ايضا موجودة بكثرة .

طبعا يوجد اضافات ربما تخص كل لغة غير الادوات التي في الاعلى ايضا ستجدها في موقع اللغة بالتاكيد و بهذا يصبح تصنيع برنامج امر طويل كل هذه اشياء نحتاجها لصنع برنامج ؟؟؟ طبعا لم يسكت المبرمجون بل راحوا يطوروا في ما يعرف ببيئة التطوير و هي عبارة عن برنامج كبير يجمع كل البرامج السابقة بل و يزيد عليها من قدرة على تنظيم ملفات المشروع مما يجعله الحل الامثل للمبرمجين . طبعا هذه البيئات تختلف باختلاف لغتك البرمجية و ربما تكون البيئة الواحدة قادرة على العمل لعدة لغات نذكر من اشهر هذه البيئات

Visual studio - Eclipsse - Net beans و غيرها الكثير

3 - واجهات المستخدم

كثير من المبرمجين في بداية مشوارهم ربما يعتقدون انهم سيصمموا اكبر البرامج و سينافسوا اكبر الشركات و ما يلبث ان يتفاجئ باول برنامج له في شاشة سوداء و يطبع عبارة Hello World الشهيرة و بعد هذا البرنامج البسيط يبدأ يعرف المبرمج قدر نفسه و يعطي البرمجة حقها لذا على المبرمج قبل ان يدخل في عالم البرمجة عليه ان يعرف كيفية بناء واجهات المستخدم في لغته

لكن ما هي اصلا واجهات المستخدم ؟؟؟؟

(Graphic User Interface)

و تختصر GUI

تعتبر واجهات المستخدم واحدة من الاساسيات المهمة في أي برنامج فالفكرة تكمن هنا في تغيير اساليب التعامل مع الحاسب و كيفية الادخال و الاخراج فبدل من تدخل القيمة في سطر الاوامر و ينفذ الحاسب العمليات عليها ثم يطبعها على الشاشة تساعد الواجهات في التعامل بسلاسة مع الكمبيوتر غير انها ساعدت كثيرا على تشغيل ملفات الصوت و الفيديو و تصفح الانترنت تخيل معي تشغيل مقطع فيديو في نظام الدوس مثلا ربما سنراه بكسل بكسل ☺

و بعد ان عرفنا الواجهات الرسومية و انها الازرار و مربعا ت النصوص و كل ما يتفاعل معه المستخدم في سبيل تسهيل التعامل مع الجهاز
انا كمبرمج كيف اصنع واجهات رسومية ؟؟؟

هنا سنضع اغلب الطرق لمعظم اللغات
و انت عزيزي القارئ عليك المتابعة في الطريقة التي تناسب مشاريعك

ملحوظة : اغلب الواجهات الرسومية تحتاج لبيئة تطوير

في C++ :

١ - دوال API : تعتبر من أقدم الطرق و أطولها إلا انها تعتبر من افضلها لما لها من قدرة عالية من الاستفادة من نظام التشغيل ويندوز و هنا نلاحظ انك تطور البرنامج دون ادوات يعني انت تصنع الادوات بنفسك و انت من يحدد خصائص الادوات بنفسك . طريقة ممتازة و قوية خاصة لتطبيقات الو يندوز لكنها ليست الافضل للمبتدئين .

٢ - Mfc (Microsoft foundation class library) : أيضا واحدة من أقدم الطرق لكنها لا زالت موجودة و يتم تطويرها هنا يوجد بعض الحركة☺ فانت تضع الادوات ثم تطبق عليها اوامر و تستخدم الاحداث مثل الضغط على زر و الكتابة و....الخ

٣ - wxWidgets : واحدة من التقنيات الحديثة المنتشرة هذه الايام و تستعملها عليها بعض الهيئات الكبيرة مثلا وكالة ناسا .

٤ - VCL (Visual component library) : أيضا ليست خيار سيء لتجربتها .

٥ - SmartWin++ : معلوماتي عنها قليلة لكنها تتميز ببناء برامج للهواتف التي تعمل بنظام windows phone .

٦ - WTL (windows template library) : هذه المكتبة نشرتها مايكروسوفت بغرض تطوير الادوات و ليس بناء واجهات .

٧ - Qt : طبعا بلا منازع الخيار الافضل حاليا لتطوير واجهات رسومية باعلى جودة ممكنة و نصيحة لكل مبرمج ان يحاول الانتقال Qt فهذه الي تدعم عدد كبير من منصات التشغيل و مع امكانية ربطها بقواعد البيانات .

نذكر من البرامج المصممة بها :

Skype - Maya - Google earth - Ubuntou

في Java :

١ - SWT (Standard Widget Tool) : تعتبر اشهر تقنية لتطوير واجهات المستخدم .

٢ - Swing : أيضا واحدة من اجمل التقنيات و تتمتع بحرية كبيرة للمبرمج لتخصيص ادواته لحصول على افضل مظهر.

٣ - AWT (Abstract window toolkit) : تعتبر اول تقنية لتطوير واجهات مستخدم منذ ظهور JAVA عام 1995.

في Python :

pyQt : كما في ال++c هنا ايضا القوة و السرعة في الانتاجية الربط بقواعد البيانات و المزيد من المميزات

اشهر البرامج التي تستعمله Ninja-IDE .

في perl : TK .

في delphi : و هي اللغة المظلومة ارى انها من اقوى اللغات و من افضلها في تطوير البرامج ذات الواجهات الرسومية مع ذلك لا تذكر كثيرا في ظل نفوذ ++c الواسع ☺ من أشهر البرامج التي بنيت بها

Kasparsky antivirous - Dev c++

طيب ذكرنا في الاعلى اغلب التقنيات المعروفة لتطوير واجهات

في هذا الجزء سنذكر بعض التقنيات الاخرى الحديثة

١ - WPF : تعتبر النقلة النوعية في البرمجة كما يقولون حيث سيصبح من الممكن تحويل البرنامج من سطح المكتب للعمل على الويب دون الحاجة للتعديل على البرنامج .

٢ - بيئة .Net : أحدث تقنية ظهرت في عالم البرمجة حيث جعلت من الممكن ان تبني برنامج بعدة لغات طالما جميعها

تتبع لبيئة .net مع كل المميزات الكبيرة التي في هذه البيئة و أهمها هو امكانية تطوير المواقع بنفس لغات البرمجة يعني اصبح تطوير المواقع هو نفسه تطوير البرامج تقول الشركة ان لكل واحدة منهم مميزات .

VB.net - C# - C++.net - F# - J# - ASP.net - Delphi.net

طبعا نجد ان اول ثلاثة متشابهات اما الباقي ففيها بعض الفروق مثلا F# لم تلقى شعبية و اظن انه هناك كتاب واحد باللغة العربية فقط يشرحها بالرغم من انها تحوي الكثير من التطويرات .

٣ - XUL : تقنية طورتها شركة Mozila من اجل بناء برامجها و نذكر ان هذه اللغة جعلت تطوير الواجهات الرسومية امر يعتمد على كتابة اكواد اشبه بتقنية API لكن مع الأسف هذه التقنية تطلب وجود برنامج فايرفوكس المتصفح الشهير عند المستخدم كي تعمل البرامج التي نصنعها و في حالة عدم توافره فلن تعمل لذا اقتصرت هذه اللغة مع الأسف على بناء اضافات لمتصفح فايرفوكس .

٤ - البرامج مفتوحة المصدر

تعتبر البرامج مفتوحة المصدر واحدة من اكثر المواضيع انتشارا في الاواسط البرمجية لما لها من مميزات تمكن المبرمج من الاستفادة من برامج سابقة في تطوير برامجه .طبعا هذه العملية ليست بالبساطة التي نتخيلها فالامر يحتاج لخبرة كبيرة في لغة برمجة (التي كتب بها البرنامج) و قدرة على تحليل الاكواد بعدها يستطيع المبرمج ان يبدأ في بناء البرامج مفتوحة المصدر .

طيب بعد قراءة هذه المقدمة بالتأكيد لديك عزيزي القارئ ثلاث اسئلة مهمة :

لماذا هذه البرامج مفتوحة المصدر ؟

من أين احصل على اكواد البرامج مفتوحة المصدر ؟

بعد أن احصل على الاكواد كيف اعدل عليها ؟؟؟☺

بالنسبة للسؤال الاول :

البرامج مفتوحة المصدر ظهرت كحركة مضادة لسياسة مايكروسوفت التي تعتمد على احتكار البرامج و المصادر المغلقةو كانت بدايات المصادر مفتوحة المصدر مع مشروع GNU و تطوير نظام لينكس الشهير و الذي تلاه انتشار اتفاقية GPL الشهيرة و هي اتفاقية التطبيقات مفتوحة المصدر .

بالنسبة للسؤال الثاني :

للحصول على اكواد مصدرية للبرامج فعليك بالتسجيل في موقع GitHub و صرح كبير عادة ما ينشر المبرمجين اكواد برامجهم عليه . كما يمكنك ان تتواصل هناك مع من يساعدوك في تطوير برامجك و توجيه النصيحة لك.

بالنسبة للسؤال الثالث :

فهو موضوع يعتمد على المصدر الذي حملته فإن اوضح صاحب البرنامج البيئة الذي استخدمها في برنامجه فعليك بتحميلها فاذا لم تتمكن من الحصول عليها فعليك ان تعتمد على البيئة التي تستخدمها طالما انها لنفس اللغة . لكن مشكلتك الحقيقية لا تكمن فقط في الادوات بل في فهم المصدر الذي بين يديك و على ما اعتقد هذه من الصعب قليلا ان يتم شرحها في كتاب .

5 - الهندسة العكسية

ربما جميعنا هذه الايام نسمع عن هذا المصطلح الغريب و يتضح لدا اكثرنا من هذا المصطلح انه علم دراسة تكوين المنتجات الهندسية . هذا التعريف بشكل عام و منه نتحدث عن أحد هذه المنتجات و هي البرامج الخاصة بالحاسب فنتيجة سياسة البرامج مغلقة المصدر و عدم قدرة الكثير من الناس على شراء البرامج أي كان نوعها او حتى العاب بدأ هذا الاتجاه الجديد في البرمجة بالانتشار بين اواسط المبرمجين . يعتبر الكثيرين ان هذا أمر غير جائز و حرام شرعا و مع ذلك يرى المهندسون العكسيون انهم يفعلون الخير لاسعاد البشرية ☺ .

نتخلص مما سبق الاستخدامات التالية :
1 - تحويل البرامج المدفوعة لمجانية .
2 - معرفة تكوين الفايروسات و تحليلها .

و هذه الأخيرة تستخدمها برامج مضادات الفايروسات كثيرا لمحاربة الفايروسات و برامج التجسس الخبيثة .

متطلبات الهندسة العكسية :

نحن لن نتطرق هنا لكيفية كسر حماية البرامج او تحليل الفايروسات لكننا سنذكر بعض الامور الاساسية .

1 - لغة Assembly أو كما يسميها البعض لغة التجميع و هي تعد اقرب لغة للغة الآلة و جميع البرامج التي نصنعها تحول في النهاية لكود بلغة التجميع منه للغة الآلة . ربما يعتقد معظم الجميع ان لغة التجميع معقدة او صعبة او انها بحاجة لعلماء لفهمها لكن الصحيح عكس ذلك فلغة التجميع في حد ذاتها سهلة في العلم و التطبيق المشكلة تكمن في قيم الذاكرة كيف تنظم قيم برامجك و تتعامل مع المسجلات التي تحمل القيم خاصة في البرامج الكبيرة و تزداد المشكلة عندما يكون الكود لبرنامج نحاول كسر حمايته . لكن كل شيء مع كثرة التجريب باذن الله يحل اهم ما في الموضوع الاصرار .

2 - انظمة العد المختلفة و هي اول موضوع ناقشناه في هذه السلسلة في الجزء الاول (شكلنا نسينا الجزء الاول ☺) و طبعا بدون انظمة العد فلن تتمكن من البدأ اصلا في الهندسة العكسية لان جميع قيم البرامج تظهر بقيم ثنائية او سداسية عشرية

هذا بالنسبة للجزء النظري طيب ماذا عن الجزء العملي ☺ ؟؟؟

تعتبر الهنسة العكسية اكثر فرع يحتاج لادوات منها ما هو لمعرفة اللغات التي برمجة بها البرامج و منها لما هو لتحويل كود اللغة للغة الاسمبلي و منها ما هو لتصنيع ملفات الكيجن و الكراك و.....الخ.

بعض أشهر الادوات في الهنسة العكسية :

١ - Ollydbg : البرنامج المسؤول عن اظهار اكواد أي برنامج بالاسمبلي .
٢ - PiED : مسؤول عن معرفة اللغة المستخدمة في صنع البرنامج .
٣ - HexEditor : يمكنك من التعديل في كود البرنامج بالاسمبلي .

يعني هذه هي أشهر الادوات . و نكتفي بهذا القدر فهذا الكتاب الغرض منه اعطاء نبذة عن كل موضوع و اترك لك العنان كي تبدع فيها او في ما تشاء منها . خلاصته لا تتوقع مني اني ساشرح كل هذه المواضيع في كتاب واحد بالتفصيل (و كمان بالمجان ☺) أرجوا ان يكون الجزء الثاني قد نال اعجابكم و ان تكونوا قد استفدتم منه و ان شاء الله مستقبلا قد نتمكن من تأليف كتب منفصلة عن هذه المواضيع .

و الى اللقاء في الحلقة القادمة !!!

اقصد الجزء القادم ☺

لجميع الاستفسارات ، التعليقات ، الملاحظات ، حالة وجود اخطاء (اشك في ذلك ☺) ارجوا مراسلتنا فورا على الايميل الخاص بي أو صفحتي على الفيس بوك .

و السلام عليكم ورحمة الله و بركاته

أخوكم

م / محمد يوسف محمد

Modi401@hotmail.com

أو

https://www.facebook.com/mohamed.yossef.583

AGASHE

2014 / 2015