المقدمة
في
تحليل و تصميم أنظمة التشغيل

للمطور
أحمد لكسايس

المقدمة

في

تحليل و تصميم أنظمة التشغيل

للمطور

أحمد لكسايس

هام

الفهرس

إهداء

هذا العمل المتواضع إهداء لابي و امي و اخي و للروح الطاهرة لأختي في الله بثينة اللبادي ، و لأصدقائي المخترع عبد الله شقرون و الخبير الأمني عبد الحميد و لأستاذي ياسين عطية و أصدقائي الذين دعموني و لكافة مطوري الأنظمة بالعالم العربي ....

عن الكاتب

أحمد لكسايس

17 سنة مطور أنظمة و خبير في الامن المعلوماتي من مواليد قرية إدويران نواحي مدينة امنتانوت بالمغرب ، رئيس قسم المطورين بشركة جارفيس المسؤولة عن نظام جارفيس المعتمد على الذكاء الصناعي مهووس بتحليل الأنظمة و محاكاتها .

للتواصل : lekssaysahmed@gmail.com

twitter.com/Lekssays

facebook.com/lekssays

www.jarviscorp.com

مقدمة

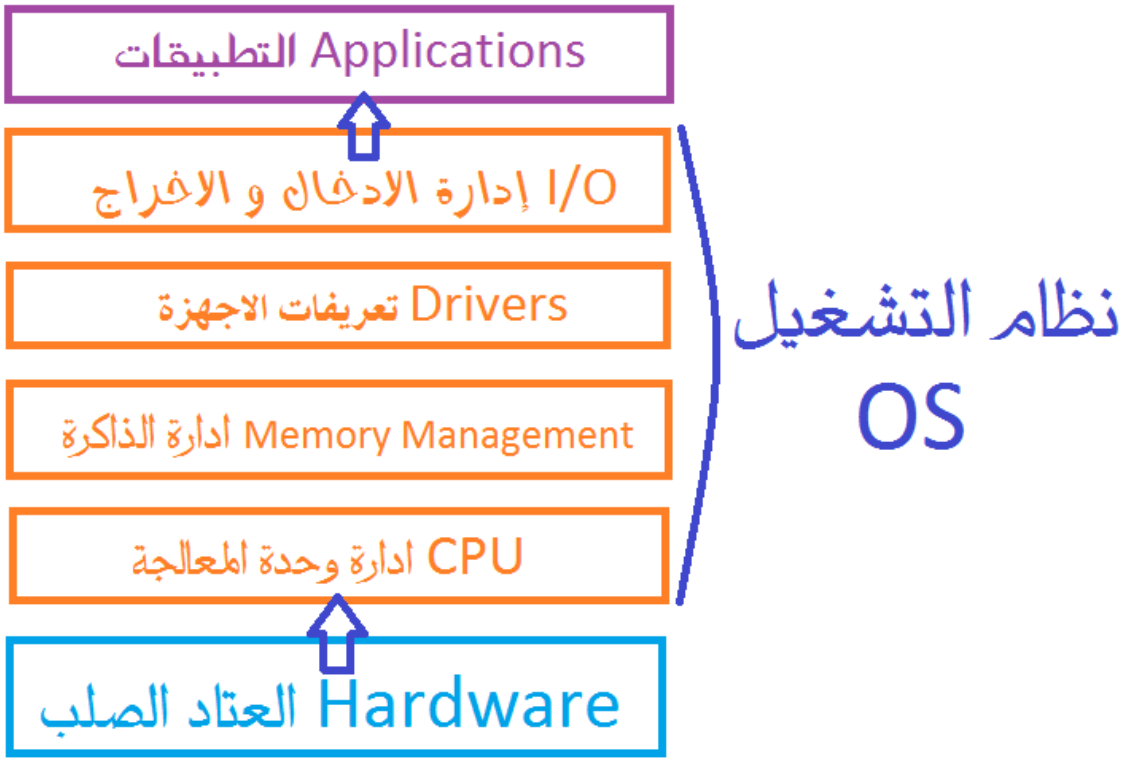
ياتي هذا الكتاب المتواضع ضمن سيرورة اثراء المحتوى المعلوماتي العربي فهو نفض غبار على المفاهيم الأساسية و على مجال تحليل الأنظمة و مقدمة لبحر واسع و قراءته رسم لخارطة الطريق للسير قدما نحو بناء أنظمة عربية متكاملة لاحتوائه على معارف أساسية في مجال نظم التشغيل و كذا بنيتها و كيفية حمايتها و نتمنى ان يكون سراجا منيرا لمطوري الأنظمة العرب ...

و الله المعين

الباب الأول : أنظمة التشغيل عبر التاريخ المعلوماتي

- ما هو نظام تشغيل ؟

نظام تشغيل او ما يطلق عليه بالإنجليزية Operating System و اختصارا OS فهو ببساطة مجموعة من البرمجيات المتكاملة التي تمكن من ربط المستخدم بالعتاد الصلب سواء كان حاسوبا ، هاتفا ...الخ ، و يقوم بمهام أساسية كإدراة مصادر الحاسوب (القرص الصلب ، الذاكرة ...الخ) و كذا إدارة الشبكات و الملفات و التحكم في مصادر الادخال و الإخراج I/O .



رسم مبسط لدور نظام التشغيل

مهام نظام التشغيل

- اعداد الحاسوب لبدء التشغيل
- إدارة وحدة المعالجة المركزية CPU جدولة المهام و الربط بين المعالجات في حالة تعددها.
- إدارة التطبيقات
- إدارة الذاكرة (الحية RAM و القراءة فقط ROM)
- إدارة وحدات الادخال و الإخراج I/O
- ربط المستخدم بواجهة رسومية GUI

لغات الحاسوب :

يمكن الفرز بين اربع أجيال من لغة الحاسوب :

- الجيل الأول : لغة الالة Machine Programming Language و هي مختلف البرامج التي يمكن برمجتها باستخدام الاكواد الثنائية .
- الجيل الثاني : و هي برامج تتم كتابتها بلغة المعالج و تسمى اللغة المجمعة Assembly Language و يتم استخدام Assembler لترجمتها للغة الالة.
- الجيل الثالث : اللغات العالية المستوى High Level Languages و هي لغات اقرب للإنسان كلغة C و Basic و تستخدم المترجمات Compilers لتحويلها الى لغة الالة .
- الجيل الرابع : و هي حزم متخصصة ك JAVA و C++ .

- ما أنواعه ؟

بالنسبة بتحديد أنواع أنظمة التشغيل فتبدو مهمة صعبة للغاية لذا تم الاجماع على تحديد مقاييس لتصنيفها و ذلك حسب :

- حسب المستخدم :

و ينقسم هذا الصنف الى نوعين (SUI (Single User Interface و تعني انه لا يمكن للمستخدم التفاعل مع اكثر من واجهة كما هو الحال بالنسبة لنظام Unix و اما النوع الثاني فهو MUI (Multi User Interface) و يتيح هذا النوع للمستخدم التفاعل مع مجموعة من الواجهات التي غالبا ما تكون رسومية GUI (Graphic User Interface و تعتبر هذه الطفرة منعرجا ملسار أنظمة التشغيل و من امثلتها أنظمة Linux و Windows .

- حسب المهام :

احادي المهام Single Tasking أي انه يتعامل مع مهمة واحدة في وقد واحد او متعدد المهام Multi Tasking أي انه يتعامل مع مجموعة من المهام في وقت واحد و حسب هذه المقاييس يمككنا تحديد اربع أنواع من نظم التشغيل :

نظام وحيد المستخدم وحيد المهام Single-user Single-tasking

نظام متعدد المستخدم وحيد المهام Multi-user Single-tasking

نظام وحيد المستخدم متعدد المهام Single-user Multi-tasking

نظام متعدد المستخدم متعدد المهام Multi-user Multi-tasking

- كيف تطور هذا المجال عبر التاريخ ؟

انبثقت فكرة أنظمة التشغيل في اربعينيات القرن الماضي فقد كانت آنذاك الأجهزة بسيطة جدا و لا توجد أنظمة تشغيل و كانت صعبة و لا يمكن لاي انسان التعامل مع الالة فقد كان المستخدم هو نفسه المبرمج و منها كتابة عمليات الادخال و الإخراج كاملة و لصعوبة المهمة تم اللجوء لإنشاء مكتبة عمليات ادخال و اخراج IOCS (Input/Output Control System) و كانت هذه البداية الفعلية لبلورة فكرة انظمة التشغيل .

و في الستينيات تم تطوير أنظمة الباتش و التي كان مبدؤها يقوم على جمع المهام في مهمة واحدة مفصولة بوحدة تحكم و كان هذا التحكم يتم عبر لغة JCL (Job Control Language) و تم تطوير هذا النظام الى ان اصبح يدعم اكثر من مهمة واحدة هذا التطور صاحبه تطور في المعالجات و الذي استمر الى فترة السبعينيات مع ازدياد حاجة الحكومات و الجامعات و الشركات في التواصل و ارسال البيانات عبر الشبكات كانت الحاجة ماسة لانظمة تشغيل قوية و في تلك الفترة تم تطوير نظام اليونيكس Unix من قبل دينيس ريتشي Dennis Ritchie و نذكر انه هو مؤسس لغة البرمجة C و كان أولا نظام يبرمج بلغة عالية المستوى C لذا لم يكن في متناول المستخدم

العادي لان كانت له أوامر معقدة و قامت بعد ذلك القصة الشهيرة للينكس حيث قام Linus Torvalds ببرمجة نواة Kernel و تم دمجها بنظام Unix و ذلك لتبسيطه و جعله اكثر تفاعلا مع المستخدم و تمت تسميته بنظام Linux . و في الثمانينيات قامت الترويج لفكرة الحاسب الشخصي و التي لاقت استحسانا كبيرا من جمهور المتتبعين آنذاك و كانت شركة ابل بمؤسسيها Steve Jobs و Steve Wozniack اول من تداول هذا المصطلح ليشكلوا طفرة نوعية في مجال انظمة التشغيل . و قد كانت شركة IBM الرائدة في المجال آنذاك و كانت نقطة ضعفها افتقارها لنظام تشغيل و تمت الاستعانة ب Bill Gates فتم تطوير نظام ل Tim Patersen ثم اهدار أموال طائلة لشراء حقوقه و بعدما تم ذلك أضيفت له تعديلات بسيطة و تمت تسميته MS-DOS . و في 20 نونبر 1985 تم اصدار اول نظام نوافذ بالعالم تحت اسم Windows 1.0 و الذي كان منعرجا اخر لنظام التشغيل حيت اتى بمميزات مبهرة منها الالة الحاسبة و ايقونات البرامج و برنامج الرسام بدون الوان .. و تطور هذه الانظمة الى ان وصلت الان لنظام Windows 8.1 . و وجب التحدث عن الأنظمة التي برزت في الآونة الأخيرة ففي 2003 تمت برمجة اول نظام ذكي للهواتف تحت مسمى الاندرويد من قبل اندي روبين Andy Rubin وتم الاستحواذ على شركة اندرويد من قبل Google و تم اطلاق Android 1.0 بعد ذلك و مع تطور الهواتف الذكية ساير

النظام ذلك و وصل لحد الان للنسخة Android 4.4 KitKat دون ان ننسى نظام ابل ios الذي وصل لنسخته ios 7 . كما توجد أنظمة أخرى في طور الإنجاز منها Firefox OS و Tizen OS بالنسبة للهواتف و اللوحيات الذكية .

الباب الثاني : بنية نظم التشغيل :

- كيف يتم تشغيل نظام التشغيل ؟

نظام التشغيل كما ذكرنا هو مجموعة من البرمجيات يتم تخزينه و نقله الى الذاكرة ليقوم بالإشراف على مختلف البرامج و التطبيقات و لإدارة وحدات التخزين ، عندما نقوم بالضغط على زر الاشتغال يبدأ تحميل برنامج يوجد في ذاكرة القراءة فقط ROM يسمى IPL (Initial Program Lood) و يقوم هذا البرنامج بتفحص العتاد الصلب و التاكد من سلامته ثم تحميل النظام من القرص الصلب ثم العمل على تنفيذ أوامر المستخدم .

- مفاهيم أساسية في نظم التشغيل

ان معرفة المفاهيم او البنيات الأساسية التي يعتمد عليها كل نظام ضرورة ملحة لولوج عالم بناء و تصميم الأنظمة و يمكننا تحديد هذه البنيات من خلال نظرتين نظرة عامة و نظرة خاصة تتمثل في

نظرة مطوري اللينكس ، سنبدأ بالنظرة العامة فكل نظام يرتكز على ثلاث مفاهيم كما يلي :

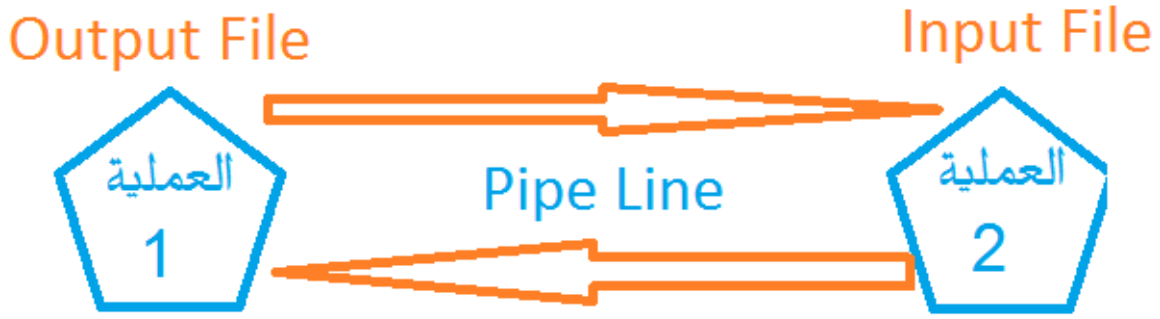
- العمل او المهمة Job / Process
- نداء النظام System Call
- المقاطعات Interrupt

فبالنسبة للمهمة Process و هو المفهوم الأكثر تداولا و تشمل قسمين من البرامج هما برنامج تحت التنفيذ Program in execution و برنامج في طور التنفيذ Program group of instruction ، و عنما يخرج البرنامج من القرص الصلب HD الى الذاكرة Memory لا يسمى برنامجا و انما مهمة او عملية Process ، و نشير الى ان في كل عملية تنفيذ لبرنامج يكون معه Process Address و ذلك لتحديد مكان العملية في المعالج ، و يمكن لعملية ان تتفرع عنها مجموعة من العمليات لتشكل شجرة عمليات تسمى اصطلاحا ب Process Hierarchical يمكن ان تكون متصلة بينها عبر قناة تسمى Pipe او Pipe Line و هي عبارة عن خيط وهمي Virtual File و تمسمى هذه العملية Inter process communication .

مثال توضيحي :

اذا توفرت لدينا عمليتان مختلفتان 1 و 2 ، وطلبت العملية 2 من 1 معلومة عبر قناة التواصل Pipe فتعتبر 1 نقطة اخراج

Output File و 2 نقطة ادخال و هذا هو مبدأ عمل Pipe و العكس .



رسم توضيحي لمبدأ عملPipe Line

و فيما يخص نداء النظام System Call هي مجموعة من الأوامر التي تخص نظام التشغيل و تخص المبرمجين خصوصا ، غير متوغرة في اللغات العالية المستوى باستثناء لغة C التي تستخدم بعض مميزات Sys call و تنقسم الى خمس أنواع :

- أوامر العمليات Job and Process Sys Call (Control) و هي عمليات مثل القص و اللصق و التشغيل و الحذف ...
- أوامر التعديل على الملف File Manipulation مثل الانشاء ، الحذف ، و الفتح و الغلق ...
- أوامر الأجهزة Devises Manipulation
- أوامر الاتصال Communication كاوامر الماوس و لوحة المفاتيح ...
- أوامر معلومات الصيانة Information maintenance و هي مرتبطة غالبا بالعتاد كالذاكرة و وحدة المعالجة المركزية ...

اما المقاطعات Interrupts فهي عملية غير متوقعة تغرير ترتيب الأوامر في المعالج مثلا : ادخال عتاد خارجي الى الحاسوب (وحدة تخزين خارجية ، Flash USB ...) و تنقسم الى ست أنواع :

- مقاطعة استدعاء المدير Execution with administrator mode و تحدث حينما يتم تثبيت برنامج جديد او تغيير اعدادات النظام و ذلك للعمل بصلاحيات المدير كمسؤول عن النظام ...
- مقاطعة الادخال و الإخراج I/O Interrupts تحدث عندما يكتمل عملية ادخال او اخراج البيانات من وحدة تخزين او في حالة وقوع خطا في العملية مثال عمليات النسخ و اللصق ...
- المقاطعات الخارجية External Interrupts او ما يسمى أيضا Software Interrupts و تحدث عندما يتم تحديد زمن محدد لانجاز عملية معينة و تتم المقاطعة فور انتهاء الوقت المحدث مثال : مقاطعة الطابعة ..
- مقاطعات الاستئناف Resume Interrupts و تعمل عندما تكون العملية في وضع الاستعداد Ready Mode .
- مقاطعات تدقيق البرامج و تحدث في ثلاث حالات هي مشاكل القسمة على 0 او محاولة تنفيذ شفرة لعملية غير صحيحة او محاولة الرجوع الى مكان في الذاكرة غير موجود .
- مقاطعات تدقيق الالة Machine Accuracy Interrupts و تحدث عندما يكون الخلل في العتاد الصلب .

و مع كل مقاطعة تشتغل الية تسمى ب ISR (Interrupt Service Routine) و هي شفرة تحدد دور كل عملية و تقوم بتنظيم المقاطعات على مستوى وحدة المعالجة المركزية CPU و ذلك بتحويل التحكم الى ISR و عند وصل مقاطعتين في نفس الوقت تعمل بمبدأين الأول مبدأ الأولوية للاهم Priority Scheme و الثاني مبدأ السماح/عدم السماح Enable/Disable .

كما ذكرنا سالفا هناك نظرتين و تحدثنا عن النظرة العامة و سنتدث الان عن النظرة الخاصة و التي تمثل نظرة مطوري اللينكس ، فالنظام حسب هؤلاء يرتكز على ثلاث بنيات أساسية هي :

- النواة Kernel
- القشرية Shell
- ملفات النظام System Files

- النواة Kernel هو نواة أنظمة اللينكس و هو مجموعة من المهام لنظام التشغيل و التي يتم تحميلها من BIOS للتعرف على العتاد و تنظيم الذاكرة و وحدات الادخال و الإخراج .. و ذلك فور تشغيل الحاسوب و هي القلب النابض لأي نظام و أي خطا بسيط في برمجته قد يكلفك الكثير قد يؤدي الى انهيار النظام بالكامل .

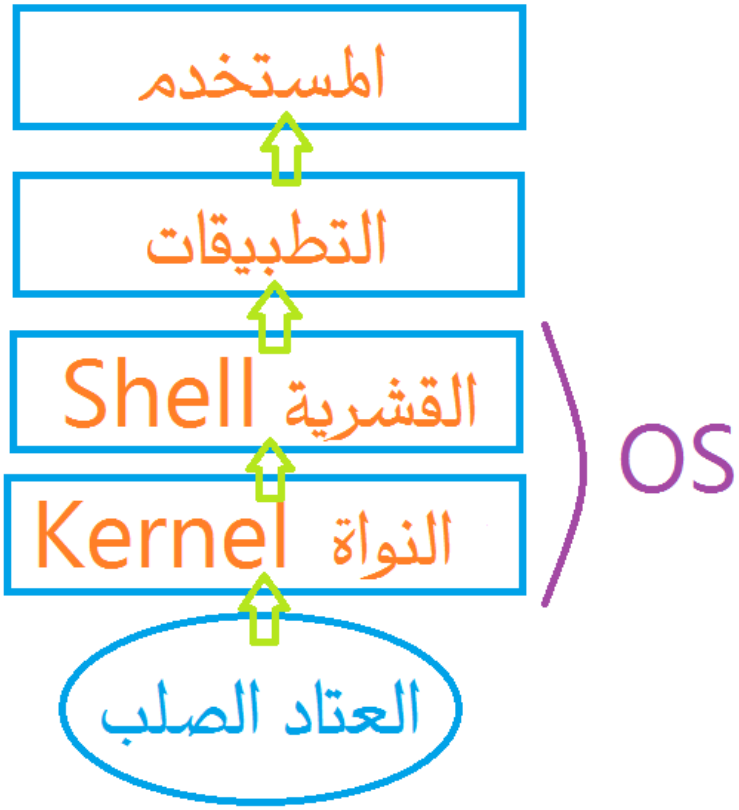
و يعد من المواضيع المتقدمة في علم الأنظمة و من المواضيع المتشعبة فيها .

و هناك نوعين من Kernel :

- نواة أحاديةMonolithic Kernel و تتميز بالاستقرار و السرعة في الأداء إضافة الى احتوائها على مجموعة هائلة من تعاريف الأجهزة و ذلك ما يخولها للتوافق مع عدد كبير من الحواسيب ، لكن رغم هذه المميزات فان لها عيوب خصوصا في برمجتها فهي معقدة البرمجة و أي خطا بسيط يختل النظام بالكامل إضافة الى كون التعديل عليها يتطلب بناء نواة جديدة.
- نواة مصغرة Micro Kernel هي نواة مقسمة للملفات و من مزاياها انه يمكن التعديل على أي جزء دون إعادة بناء النواة و لكن هذه الخاصية هي سيف ذو حدين لان هذه النواة تربط كل قسم بالاخر مما يشغل مساحة اكبر في المعالج.

و تعد نواة اللينكس Linux Kernel من اشهر النوى و اكثرها استقرارا بالإضافة الى كونها مفتوحة المصدر و يمكنكم زيارة الموقع الرسمي له من خلال الرابط : www.kernel.org

- القشرية Shell هي برمجية تعمل بالطبقة العليا من النظام و هي الواجهة التي تظهر الى المستخدم و تربطه بالعتاد الصلب و تمكنه من تطبيق ألاوامر .



رسم مبسط لدور القشرية و النواة

- ثم ملفات النظام System Files وهي الملفات التي يتم تخزينها في القرص الصلب و تعتبر أساس نظم التشغيل .

و قبل الشروع في بناء أي نظام علينا تحديد الفرق بين معمارتي 32bit و 64bit يتمثل أساسا على مستويين البرمجي و العتادي فبرمجيا يتمثل الفرق في سرعة النظام و كفاءته بالنسبة للمعالج فان كان ذا انوية واحدة Mono Core فانه سيقبل أنظمة ذات 32bit فقط و ان كان ذا انويتين او اربع او ست Dual Core or Quad Core or Six Core فانه سيقبل الأنظمة ذات 32bit و 64bit بالإضافة الى ان معالجات 32bit لا تعرف الذاكرة RAM اذ كانت اكبر من 4gb بينما معالجات 64bit فيمكنها تعرف ذاكرة و ان كانت اكبر

من 4gb و نشير الا ان هذا النوع من المعالجات لا تظهر كفاءته الا ان توفرت مواصفات عالية في الجهاز ك 4gb في الذاكرة و بطاقة رسومية ذات جودة 720p و اكثر .

و بالنسبة للفرق من الناحية العتادية فيختلف من حيث عدد الانويات و الخطوط الالكترونية Bus التي تربط بين مختلف مكونات العتاد الصلب فمثلا معالجات 32bit لها خطوط الكترونية دقيقة تسمح بمرور 32 bit فقط و بالنسبة لمعالجات 64bit لها خطوط تسمح بمرور 64 bit مما يفسر سرعة هذه المعالجات . و نشير الا ان bit هي اصغر وحدة ناقلة لمعلومة او معنى معين و تساوي 8bit = 1 bytes. فمثلا الحرف A يساوي 8bit ...

نصائح عملية لبناء نظام تشغيل متكامل :

ربما بناء نظام تشغيل متكامل يظل حلم يراود جل مطوري الأنظمة في العالم ، و من خلال تجربتي الشخصية في بناء الأنظمة و تجربتي العملية مع شركة جارفيس الذي امثل بالمناسبة مطور النواة Kernel فيه ، فاني اقدم هذه النصائح لكل من يريد تصميم نظامه الخاص من الصفر أي برمجيا دون الاعتماد على أي مصدر :

- تحديد الهدف من بناء النظام مثلا وظيفته ، ماذا يمكنه ان يقدم أي فكرة عامة حوله .

- بناء تصميم على الورق و تقسيم العمل الى اشطر كشطر بناء ملفات النظام ، شطر بناء الكيرنل و شطر ادماج التطبيقات و الواجهة الرسومية .
- تحديد الأجهزة التي يخصص لها النظام من خلال تحديد أنوية المعالجات 32bit و 64bit .
- قراءة الاكواد البرمجية لمختلف الأنظمة و محاولة فهم اليتها و ذلك بتصفح مواقع البرمجيات المفتوحة المصدر .
- التركيز على بناء نواة قوية و مستقرة للنظام و افضل لغة C في برمجتها .
- التركيز على جانب الحماية المعلوماتية و حماية بيانات النظام و كذا السهر على حماية المستخدم من المخاطر الالكترونية التي تحيط به.

و من هنا نخلص الى ان ما يضفي على النظام قيمة هي الفكرة التي جاء بها و نشير الى انه يمكن الاعتماد على نواة اللينكس التي تعتبر مفتوحة المصدر و افضل دائما النسخ المستقرة Stable لانها تكون نادرة المشاكل(راجع فقرة الكيرنل الذي ذكرناها سالفا) ، و هناك موقع يخول لنا بناء توزيعتنا لينكس من الصفر و ذلك بتوفيره لنظام قابل للتطوير و يقوم المطور ببناء الملفات و ادماج الواجهات الرسومية مثلا KDE و GNOME و كذا ترقية النواة يمكنك

مراجعته و تحميل الكود البرمجي و كتاب المستخدم و ذلك لمساعدتك على بناء نظامك و ذلك من خلال الرابط :
www.linuxfromscratch.org
و بالنسبة للمبتدئين الذين يريدون اكتشاف هذا المجال فانصحهم بموقع www.susestudio.com فهو موقع جميل يخول لك بناء توزيعتك على نظام Opensuse الذي يندرج ضمن أنظمة اللينكس ، و التعديل عليها كما شئت هذا الموقع يمكنه من ادماج ادواتك المفضلة و واجهتك الرسومية المفضلة إضافة الى اختيار الانوية التي تريد ان تشتغل عليها توزيعتك و ذلك في اقل من 30 دقيقة .
و من هنا يمكننا تحديد الفرق بين نظام تشغيل و توزيعة ، فنظام التشغيل يكون فيه المبرمج هو المطور أي تتم برمجة النواة و بناء ملفات النظام دون الاعتماد على اية شفرة مصدرية مثال اللينكس ، اليونيكس ، اماك ، الويندوز ... اما التوزيعة فتقوم على تطوير نظام تشغيل او الاعتماد على شفرة مصدرية و غالبا ما تكون مفتوحة المصدر كشيفرة اللينكس و منها انبثقت توزيعات مثل :
Ubuntu , Debian , Opensuse ...

الباب الثالث : حماية أنظمة التشغيل

يبقى جانب الحماية المعلوماتية من اهم الجوانب التي يجب التركيز عليها و التي لا يكترث لها المستخدم العربي على العموم مع ان العالم العربي يزخر بطاقات و مواهب شابة قدمت الشيء الكثير في هذا المجال و لعل افضل وسيلة للحماية من المخاطر الأمنية هي فهمها و معرفتها ثم تعلم بعض التقنيات لتجنبها .

فلتحليل نظام او اختبار مدى حمايته نرتكز على معايير او مقاييس تسمى Protection Measures و هي كالاتي :

- Dual Mode Protection و يتم ذلك بالتفريق بين عمليات النظام و عمليات المستخدم عبر إضافة Mode Bit

  و التي فإذا كانت الخانة 0 فانها للنظام و ان كانت 1 فانها للمستخدم و هذه هي لغة الحاسوب او لغة الالة لذا فبرامج النظام تسمى Monitor Mode و برامج المستخدم User Mode .

- I/O Protection و توجد دائما في Monitor Mode لان المستخدم لا يمكنه تحديد احداثيات حفظ البرنامج او العملية داخل الذاكرة و تتم في عمليات الادخال و الإخراج و تشمل جميع اوامرها.

- Memory Protection أي حماية الذاكرة و منع برامج المستخدم للولوج اليها و تعتبر المسؤولة عن تحديد المساحة في الذاكرة و يسمح لبرامج النظام الولوج فقط.
- CPU Protection أي حماية وحدة المعالجة المركزية و تتم عبر ما يسمى Timer و ذلك للفصل بين العمليات المرغوب فيها و الغير المرغوب فيها و ذلك بتحديد زمن محدد لكل عملية .

فمن المخاطر الأمنية التي تواجه الأنظمة نجد مشاكل المنتج و المستهلك Buffer فيمككنا الحديث عن هذه المشاكل عندما يتم انتاج كمية من المعلومات و لا يمكن استهلاكها و يتم تخزينها في Buffer يتم امتلاؤه و يمكننا الفصل بين نوعين :

- Unbound Buffer أي Buffer غير محدود و هذا هو الاخطر لانه يعرض النظام للتوقف الكامل .
- Bound Buffer أي Buffer محدود و يتوقف بامتلاء Buffer .

و نشير الى انه يجب التمييز بين ثغرات Buffer Overflows التي تصيب التطبيقات و البرامج و بين مشاكل Buffer التي تصيب أنظمة التشغيل فهي متشابهة من حيث المبدأ و مختلفة من حيث الاستغلال و التطبيق .

و نجد ان البرمجيات الخبيثة(الفيروسات Virus، الديدان Worms، احصنة طروادة Trojans ، Backdoors ...) تعد من اكبر المخاطر التي تهدد سلامة و خصوصية المستخدم على حد سواء لذا يجب علينا الفصل بين مختلف هذه المخاطر الأمنية و تسمية الأمور بمسمياتها :

- الفيروسات Virus هي برمجيات صغيرة لها خاصية "التكاثر" ليس بمعناها البيولوجي و انما لانها تنسخ نفسها بنفسها دون اذن من المستخدم و تقوم بالانتقال عبر وحدات التخزين الخارجية ، و لا تقتصر خطورتها عند هذا الحد و انما بإمكانها حذف الملفات الأساسية للنظام .
- الديدان Worms هي برمجيات خبيثة صغيرة تتميز بسرعة الانتشار دون الحاجة الى وسيط مادي فتستغل ثغرات الأنظمة و تنتقل عبر الشبكات و ذلك لحصد عدد كبير من الضحايا ...
- احصنة طروادة Trojans تمت تسميتها وفق الأسطورة اليونانية و هي عبارة عن برمجيات بسيطة متنكرة تاخذ صلاحيات المستخدم من الضغط عليها و يعمد المخترق الى استعمال الهندسة الاجتماعية لاقناع المستخدم بالضغط على البرمجية .

- Backdoors هو برمجية تعمل كخلفية و تقوم بفتح بورت للتحكم في جهاز المستخدم او للتجسس عليه و نقل معلوماته الشخصية من دون علمه و هذا الأسلوب هو الذي تعتمده الشركات و الحكومات للتجسس على عملائها و مواطنيها و من انواعه SSH و RAT .

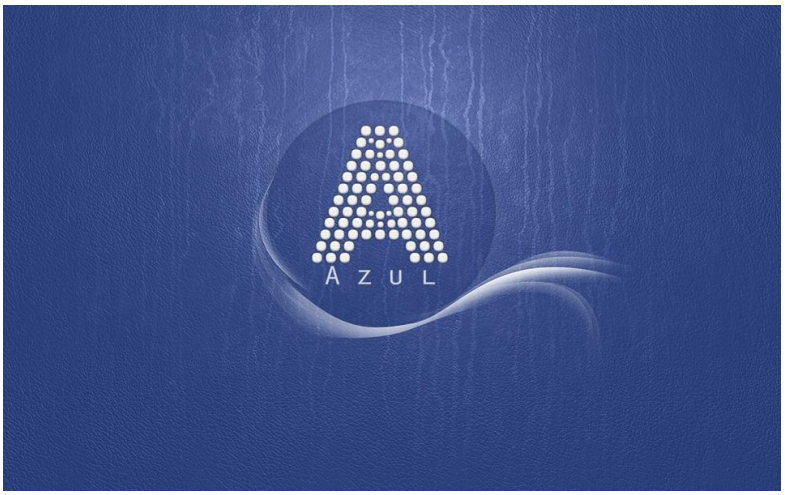
و نذكر ان هناك اخطار امنية أخرى كتجاوز كلمة المرور للنظام و التي يعاني منها نظام Windows 7 بكثرة إضافة الى نظامي Android و ios و ان من اكثر الأنظمة المعرضة لهذه المخاطر هي أنظمة الويندوز .

و لحماية النظام من هذه الاخطار يتم اعتماد اليات منها :

- جدران النار Firewalls و هي خط الدفاع الأول التي تمر منه البيانات التي تدخل للنظام او تخرج منه و يفرز بين البيانات المسموح لها بالدخول او الخروج و ذلك بمراقبة اتصالات النظام في الشبكات الداخلية و الخارجية.
- تامين المنافذ الشبكية Protection of Ports : و ذلك بمراقبة المنافذ المفتوحة في نظامك و قفل المنافذ التي تعرف بانها منافذ تستعمل للاختراقات و تكون خاصة بالبرمجيات الخبيثة و تستعملها لارسال بيانات المستخدم و تداولها .

الباب الرابع : مشاريع أنظمة تشغيل عربية مفتوحة المصدر

- نظام Azul/ ⵣⵓⵍ/ ازول : و هذه الكلمة تعني الامازغية السلام و التي تعتبر لغتنا الام و الذي نتشارك فيها انا و الأخ حمزة بوالرحيم المطور لهذه التوزيعة و عرض علي شخصيا العمل معه و قبلت لانه جاء في مرحلة طورت فيه نظامي الخاص و فترة عملي مع شركة جارفيس و اردنا مشاركة الخبرات و ذلك لادماج الذكاء الصناعي فيه لانتاج نظام امازيغي عربي متكامل و مفتوح المصدر و نعمل الان على دمج اللغة العربية و الامازغية و ذلك ليكون اول نظام عربي امازيغي يقوم على الذكاء الصناعي و يمكنكم زيارة الموقع الرسمي عبر الرابط : www.azulos.org



- نظام اعجوبة Ojuba : توزيعة مبنية على فيدورا و تعتمد افتراضيا الواجهة المكتبية Gnome و تدعم معمارتي 32bit و

64bit . و لمعلومات اكثر حول هذه التوزيعة يرجى زيارة الموقع الرسمي عبر الرابط :

www.ojuba.org

- نظام هلال Helal : توزيعة عربية مبنية على Ubuntu و تتميز بتغطيتها لاغلب احتياجات المستخدم و تسعى لتوفير نظام عربي يدعم العربية افتراضيا و تتميز بالسرعة و الكفاءة و يمكنكم زيارة موقع التوزيعة الرسمي للمزيد من المعلومات عبر الرابط :

www.helallinux.com

و لا يسعنا الا دعم المشاريع العربية و تشجيع مطوريها و ذلك للنهوض بالمستوى العربي و اعلاء مكانه في العالم الافتراضي.