

في ادارة البرامج ان شاء الله سوف نتحدث عن ثلاثة انواع من طرق تركيب البرامج و في الجزء الأول سوف نتحدث ان شاء الله
عن تثبيت البرامج من المصدر (source) و هذا الموضوع مهم جدا جدا لذلك انصحك بقراءته بتمعن و الاستفسار عن اي شيء
لا تفهمه

-من اول و اشهر طرق تركيب البرامج هي عن طريق تثبيت البرامج من المصدر بمعنى ان يتم تحويل شفرة البرنامج الى ملف تشغيل بواسطة المستخدم نفسه و هذا النوع منتشر الى الآن و يفضل الكثير نظرا لأسباب عديده نذكر منها على سبيل المثال لا الحصل

- 1- تستطيع ان تقرأ شفرة البرنامج و تتأكد من عدم وجود اي فيروسات او ثغرات امنييه بنفسك
- 2- تستطيع ان تعدل و تغير في شيفرة البرنامج و تركب فقط ما تريده من المميزات او تعدل عليها

-و طريقة تركيب البرامج من الشيفرة تختلف من برنامج الى اخر لكن الكثير من البرامج يعتمد على خطوات التركيب التالية (قليلة البرامج التي لا تعتمد على هذه الطريقة) :

1- تحميل الشفرة من الأنترنت

2- فك ضغط ملفات شفرة البرنامج

3- قراءة الملفات INSTALL README التي تأتي مع البرنامج

4- تشغيل الملف

./configure

إذا كان موجود مع ملفات البرنامج

5- تعديل الملف

Makefile

إذا كنت تحتاج لذلك

6- تحويل شفرة البرنامج الى برنامج تنفيذي بواسطة الأمر make

7- تثبيت البرنامج الى جهازك بواسطة الامر

make install

-لنفترض اننا قمنا مثلا بتحميل الملف

support-bot.tar.gz

من الأنترنت فأننا نتبع التالي لتركيبه

اولا نكتب الأمر التالي لفك ضغط الملف

gunzip support-bot.tar.gz

(كتبنا الأمر `gunzip` فقط لان الملف ينتهي بالامتداد `gz` ، و سوف نشرح هذا الأمر في وقته ان شاء الله)
(لو كان الملف ينتهي بـ `bz2` بمعنى يكون اسمه `support-bot.tar.bz2` فأننا نكتب `bzip2 support-bot.tar.bz2`)

بعد ذلك سوف يتكون لدينا ملف بالأسم

```
support-bot.tar
```

لذلك نكتب الأمر التالي لفك اشفة الملفات

```
tar xvf support-bot.tar
```

(استخدمنا هذا الأمر لان الملف ينتهي بـ `tar` و سوف نشرح هذا الامر في وقته ان شاء الله)

الآن سوف تلاحظ انه يعرض لك اسطر عديده و اسماء الملفات التي يقوم بفك اشفتها ثم سوف تجد مجلد بالأسم
`support-bot/`

- الخطوة التالية هي الدخول الى المجلد بواسطة الأمر

```
cd support-bot
```

- بعد ذلك نقوم بقراءة الملفين `README` و `INSTALL` بواسطة الأمرين

```
more INSTALL  
more README
```

حيث ان هذين الملفين مهمين

- بعد ذلك نقوم الآن بتشغيل الملف `configure` اذا كان موجود (تأكد من انك قراءة الملفين `README` `INSTALL` فقد تكون طريقة التركيب مختلفة)
و تشغيل الملف `configure` هو بكتابة التالي

```
./configure
```

(هذا الأمر يقوم بجمع معلومات عن جهازك و نظامك و يتأكد من توفر جميع المتطلبات لكي يعمل البرنامج الذي نقوم بتركيبه)
سوف تظهر لك اسطر عديده و معلومات عديده

- الآن اذا سارت العملية بنجاح و لم تحدث اي مشكلة اثناء تشغيل الملف

```
./configure
```

فسوف تجد ملف بالأسم

```
Makefile
```

افتح هذا الملف و قم بأي تعديل تريد به حيث ان هذا الملف سوف يحتوي على اعدادات تحويل الشفرة الى برنامج و تشبيته (لا تفعل هذه الخطوة سوى اذا كنت فعلا تحتاج لعمل شيء معين في الملف)

و اليك بعض الأعدادات الشائعة في اغلب البرامج و التي قد تحتاج لتعديلها

```
prefix = /usr/local
```

هذا يحدد المكان الذي سوف يتم تركيب البرنامج به

`bindir = ${exec_prefix}/bin`

يحدد المجلد الذي سوف يتم تركيب ملفات البرنامج التشغيلية به

`maindir = ${prefix}/man`

يحدد المكان الذي سوف يتم به تركيب ملفات المساعدة الخاصة بالبرنامج

- بعد ان تنتهي اكتب التالي

`make`

(هذا الأمر سوف يقوم بتحويل شفرة البرنامج الى ملفات تشغيلية تستطيع بعدها استخدام البرنامج)

ملاحظه : لو حصل و ظهرت سطر مشابه للتالي

`make: *** No targets specified and no makefile found. Stop.`

فهذه عبارة يخبرك بها البرنامج `make` انه لم يجد اي ملف بالاسم `Makefile` في المجلد الذي كتبت به الأمر و في اغلب الحالات يكون ذلك بسبب ان الأمر `configure` قد توقف عن العمل اثناء تشغيله و لم يكون الملف `Makefile` لانه ربما اكتشف اما ان برامج معينه غير متوفره لديك او ان هذه البرامج متوفره لديك و لكن اصدار قديم و في هذه الحالة تحتاج لإعادة تنفيذ الأمر `configure` و قراءة اي رسائل خطأ قد تصدر منه

- بعد ان تنتهي من الأمر `make` عليك الآن كتابة التالي

`make install`

حيث ان هذا الأمر يقوم بتهيئة البرنامج في نظامك كي تستطيع استخدامه

و بذلك تكون قد انتهيت من تركيب البرنامج

الدرس سوف يكون بخصوص المكتبات و ماهي المكتبات و كيفية التعرف على المكتبات المطلوبه لتركيب أي برنامج أو تشغيله

- عندما يقوم مبرمج ما بكتابة برنامج فإنه في أغلب الأحوال لا يقوم بكتابة المهام الشائعة مثل كتابة كود يعرض النصوص على الشاشة او كود مثلا يقوم بتقبل الأحرف من لوحة المفاتيح و لكن يستعين بملفات تكون جاهزه و تحتوي على هذه الأكواد لكي لا يضطر هو ولا غيره من المبرمجين من كتابة هذه الأكواد في كل مره يريد بها عمل برنامج ما ، هذه الملفات تسمى بالمكتبات حيث انها تحتوي على مجموعه من المهام الشائعة التي يطلبها العديد من البرامج و تكون جاهزه للعمل في حال طلب منها اي برنامج مهما معينه

توجد العديد من المكتبات و ليست محددة بعدد معين من المكتبات فتوجد مثلا مكتبات للصور و مكتبات للصوت و يستطيع تقريبا اي مبرمج ان يصنع مكتبه خاصهه او حتى نشرها على الأنترنت لكي ستفيد منها الآخرين



- لكي يتم بناء برنامج معين و لكي يعمل فإنه يحتاج للمكتبات التي تم كتابته ليستعملها
- بعض البرامج أثناء بناء شفرتها يتم إدراج المكتبات المطلوبه في ملف البرنامج النهائي و بذلك لا يحتاج البرنامج إلى المكتبه لكي يشتغل و يسمى هذا النوع بـ **Statically compiled application**
- البرامج الأخرى تقوم فقط أثناء بناء البرنامج من شفرته بربط ملف البرنامج النهائي بالمكتبات التي يحتاجها و بذلك يحتاج البرنامج اثناء تشغيله إلى المكتبات لكي يعمل و يسمى هذا النوع بـ **Dynamically compiled application**
- بناء شفرات البرامج **Dynamically** أفضل بكثير من بناء شفرات البرنامج **Statically** و السبب هو انه في حالة بناء شفرة البرنامج **Statically** سوف يكون ملف البرنامج النهائي أكبر في الحجم و أيضا تخيل انك شغلت اكثر من برنامج من نوع **Statically** فذلك سوف يستهلك ذاكر من جهازك أكبر لانه توجد اجزاء من كل برنامج **Statically** مشغل متشابهه (وهي المكتبات التي تم دمجها) و يتم حجز لكل منها مساحه خاصه بها لكن في حالة البرامج التي تكون **Dynamically** فإنها تستهلك مساحه واحده فقط للمكتبات التي يشترك بها اكثر من برنامج يعمل (اسف لم اجد اي طريقه اوضح بها اكثر من هذه الطريقه)
- المكتبات تخزن في لينوكس في أكثر من مجلد و الرئيسية و المهمه منها هي :

/lib
/usr/lib
/usr/X11R6/lib

- أسماء ملفات المكتبات في لينوكس يكون على الشكل التالي

libraryname-major-minor-patch.so

لاحظ ان أمتداد الملف هو **so** وهو اختصار لـ **Shared Object** ، و كمثل تفصل فيه معنى او هيئة اسم ملفات المكتبات دعنا نأخذ على سبيل المثال المكتبه

libcrypt-2-1-3.so

وهي الأسم السابق يدل على :

أسم المكتبه هو : **libcrypt**

الإصدار الرئيسي هو : ٢

الإصدار الفرعي هو : ١

مستوى الرقععه هو : ٣

- سوف تجد في جهازك ايضا بعض المكتبات اسماء ملفاتها مكتوبه بالشكل

libraryname.so
libraryname.so.major

و كمثل على مكتبتنا السابقة فقد تجدها بالاسم

libcrypt.so
libcrypt.so.2

و في الحقيقة هذه مجرد ارتباط يشير لنفس هيئة الملفات التي تحدثنا عنها

libraryname-major-minor-patch.so

و تم وضع هذه الأرتباطات بهذه الأسماء كي لا تحتاج في كل مره تعمل ترقيته لأحد المكتبات ان تقوم بإعادة بناء جميع البرامج التي تعتمد عليها كي تعرف هذه البرامج اسم المكتبه الجديد

التعرف على المكتبات المطلوبه لعمل برنامج معين

اذا كان لديك برنامج معين و تود التعرف على ماهي المكتبات المشتركه التي يستخدمها كل ما عليك هو استخدام الأمر ldd

يتبعه ملف البرنامج

و كمثل افرض اننا نريد ان نعرف المكتبات المطلوبه لكي يعمل البرنامج wget فإننا نكتب

```
[root@localhost src]# ldd /usr/bin/wget  
libc.so.6 => /lib/i686/libc.so.6 (0x42000000)  
/lib/ld-linux.so.2 => /lib/ld-linux.so.2 (0x40000000)
```

وهذا يعني ان البرنامج wget يستخدم المكتبه libc الأصدار السادس و المكتبه ld-linux الإصدار الثاني

إضافة مسارات المجلدات

عندما لا يجد اي برنامج احد المكتبات المشتركه التي ستخدمها سوف يعرض رسالة خطأ ثم يقوم بإغلاق نفسه فإذا كنت تعلم ان هذه المكتبه موجوده بالفعل لديك لكن ليست موجوده في مجلدات المكتبات الافتراضيه (راجع المجلدات الافتراضيه في الأعلى) فستطيع ببساطه كتابة الأمر

```
export LD_LIBRARY_PATH=/to/the/new/path
```

بحيث تستبدل to/the/new/path/ بمسار المكتبه

إعداد المكتبات المشتركه الجديدة

عندما تقوم بتهيئة مكتبات جديده إلى جهازك بنفسك (اقصد بدون استخدام احد برامج الإدارة مثل rpm) فإنك تحتاج بنفسك ان تبلغ النظام عن هذه المكتبات الجديده (حتى لو وضعتها في المجلد الافتراضي للمكتبات) و لكي تخبر النظام عن المكتبات الجديده كل ما عليك هو كتابة الأمر

```
ldconfig
```

اما اذا كانت المكتبات في مجلد ليس احد المجلدات الافتراضيه فإنك تحتاج لإضافة مسار المجلد في الملف

/etc/ld.so.conf

ثم تكتب الأمر

ldconfig

الذي يقوم بدوره بإنشاء و تحديث الملف

/etc/ld.so.cache