**نظام التشغيل ويندوز**

**🔹 تعريف نظام ويندوز**

ويندوز هو **نظام تشغيل** من إنتاج مايكروسوفت، يقوم بإدارة **البرامج** و**المستخدمين** و**العتاد**، ويوفّر **واجهة رسومية سهلة** للتفاعل مع الحاسوب.

**🔹 مكونات النظام (من الأعلى إلى الأسفل)**

1. **المستخدم (User)**  
   يُعطي الأوامر عبر الماوس أو لوحة المفاتيح (مثل فتح ملف أو تشغيل تطبيق).
2. **البرامج (Applications)**  
   تنفّذ أوامر المستخدم، وتطلب من نظام التشغيل الوصول إلى الموارد (ملفات، ذاكرة، شبكة...).
3. **نظام التشغيل (Windows OS)**  
   يترجم أوامر البرامج إلى تعليمات يمكن للعتاد فهمها، ويتحكم في إدارة الذاكرة، المعالج، الملفات، والأمن.
4. **السائقون (Device Drivers)**  
   هي **برامج خاصة تربط نظام التشغيل بالأجهزة المادية**.  
   يقوم السائق بدور **الوسيط** بين النظام والعتاد، فيحوّل الأوامر العامة من ويندوز إلى تعليمات محددة يفهمها الجهاز نفسه (مثل الطابعة، كرت الشاشة، أو القرص الصلب).
5. **العتاد (Hardware)**  
   المكونات الفيزيائية للحاسوب مثل **المعالج (CPU)**، **الذاكرة (RAM)**، **القرص الصلب**، **الشاشة**، و**لوحة المفاتيح**.

**🔹 كيف تتم العملية**

1. المستخدم يطلب تنفيذ أمر (مثلاً فتح صورة).
2. البرنامج يرسل الطلب إلى نظام التشغيل.
3. نظام التشغيل يمرّر الطلب إلى **السائق المناسب** للجهاز المطلوب (مثل سائق القرص).
4. السائق يرسل الأوامر الفعلية إلى العتاد وينتظر النتيجة.
5. العتاد ينفّذ العملية ويرسل الاستجابة.
6. النظام يعيد النتيجة إلى البرنامج → والبرنامج يعرضها للمستخدم.

**🔹 باختصار شديد**

المستخدم ↔ البرنامج ↔ نظام التشغيل ↔ **السائق** ↔ العتاد

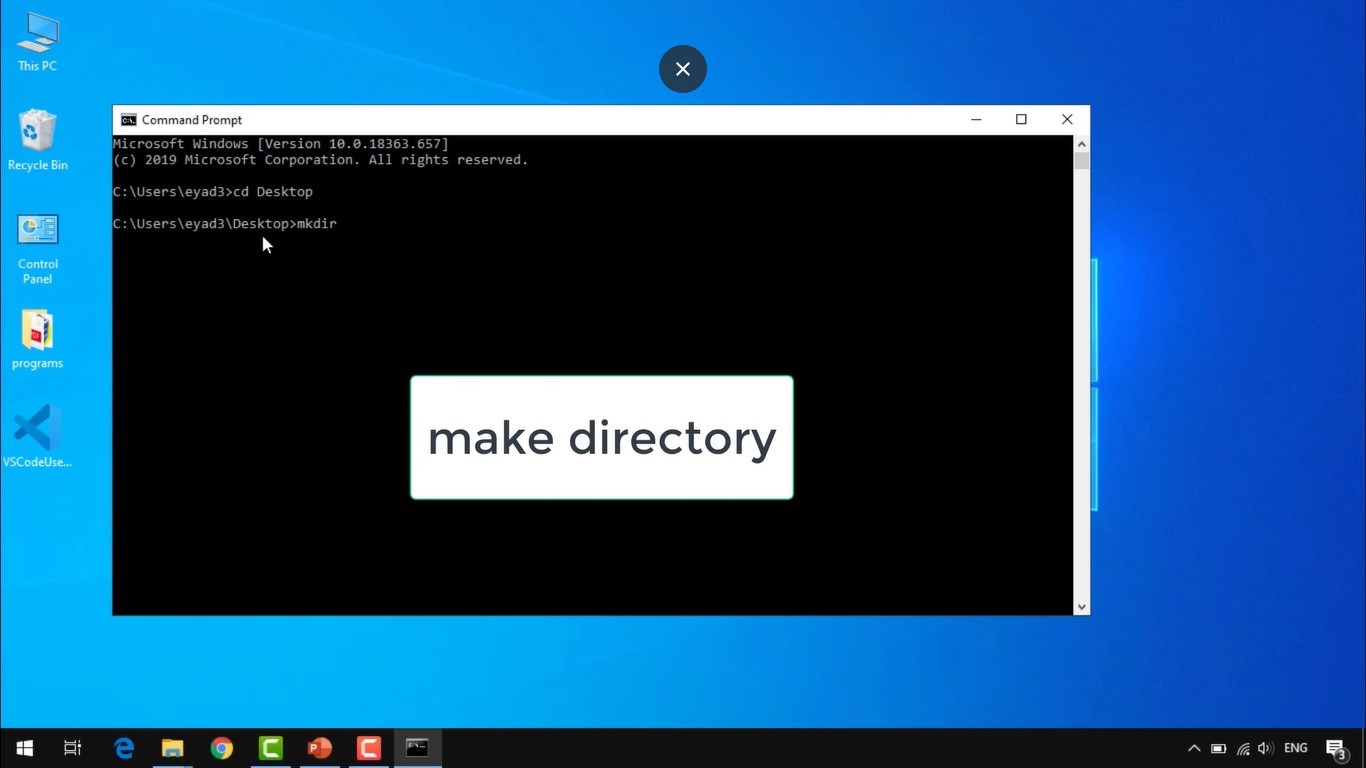
cd <folder\_directory>

نضيف مكان <folder\_directory> مسار العمل.

لأنشاء مجلد في ويندوز وماك نتسخدم :

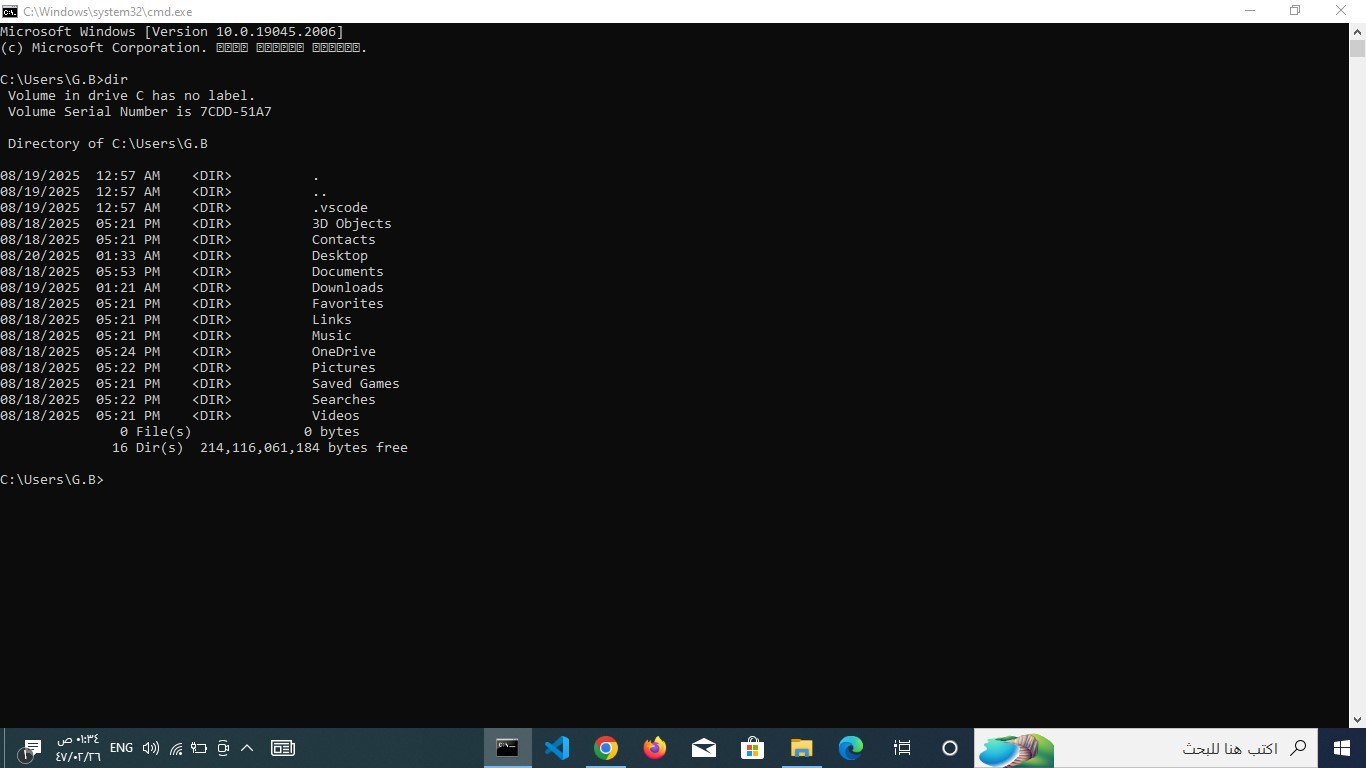
mkdir <folder\_name>

نضيف مكان <folder\_name> اسم المجلد الجديد.



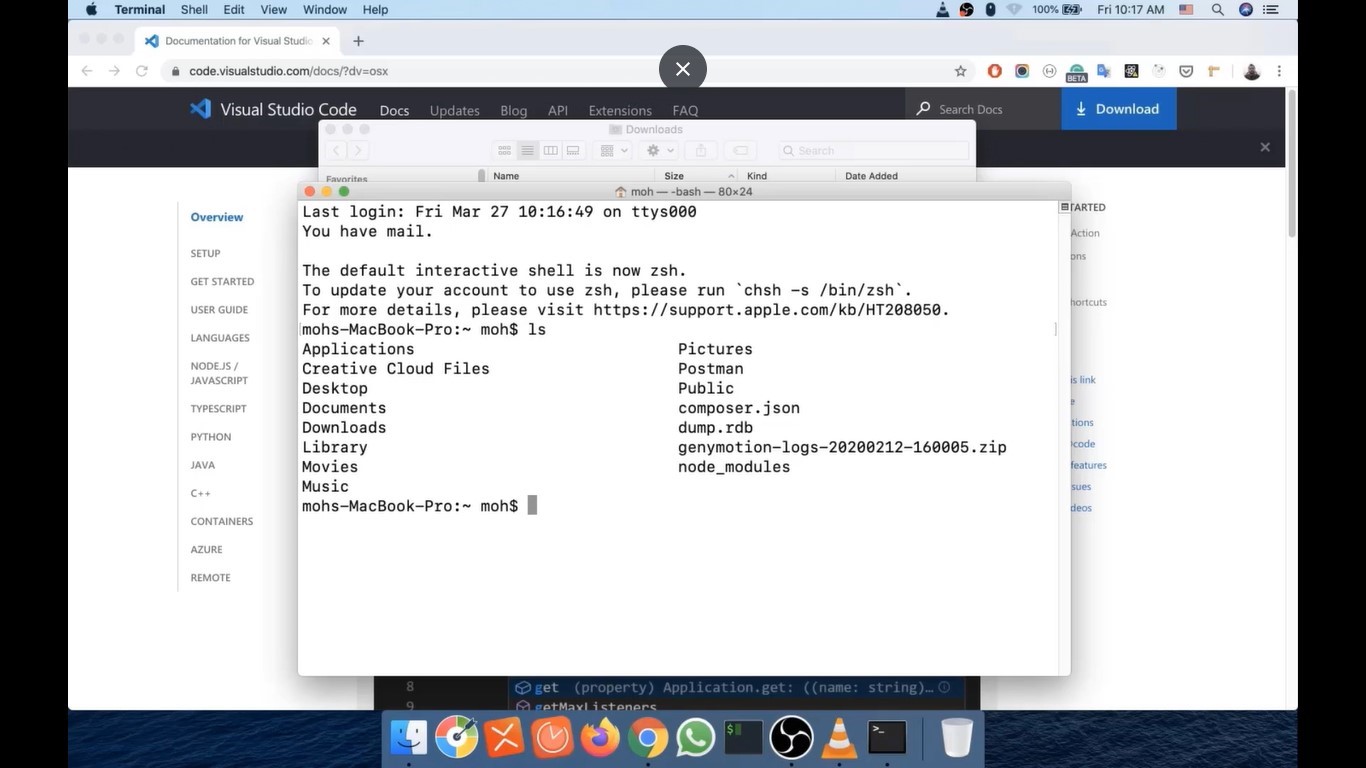
إظهار الملفات والمجلدات الموجودة ضمن المسار الحالي:

dir



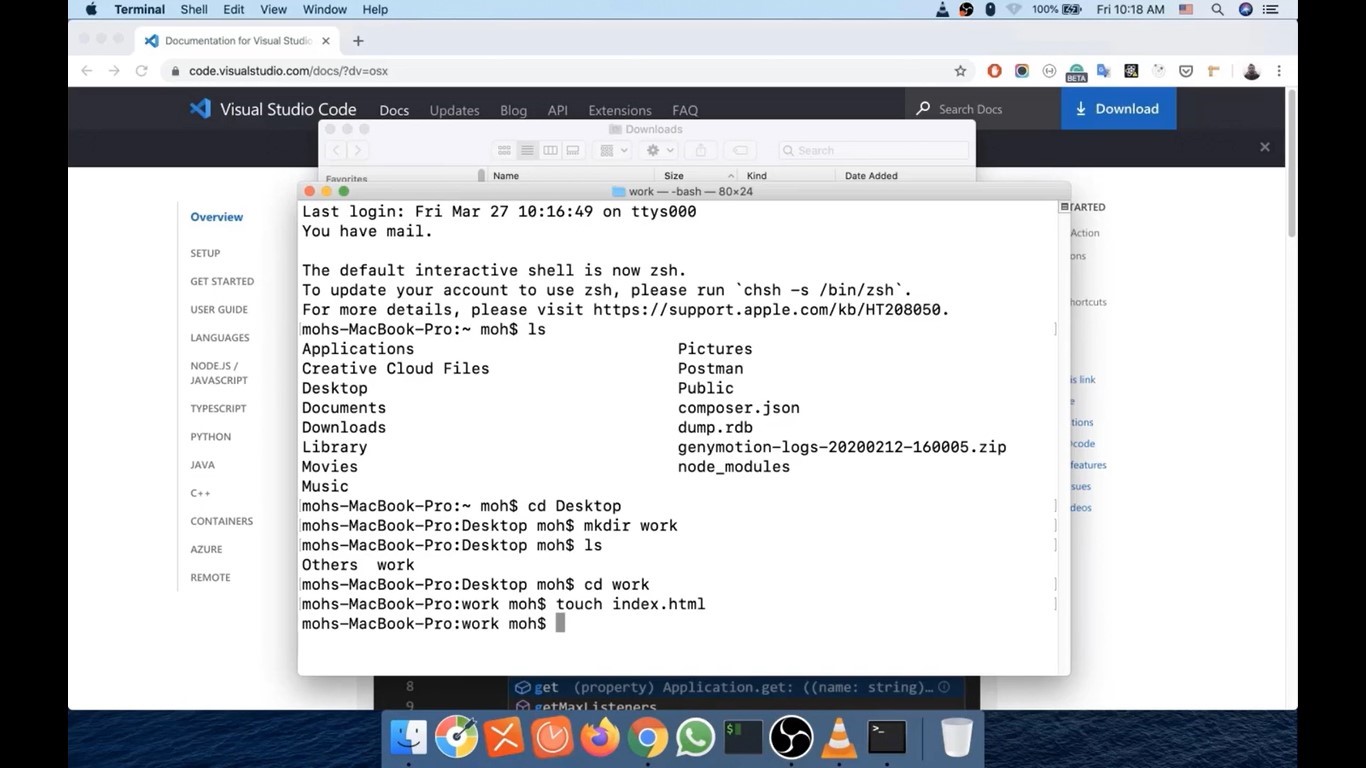
عرض قائمة بالملفات ضمن مسار المجلد:

ls



* لإنشاء ملف جديد :
* touch <file\_name>.<extension>

نضيف مكان <file\_name> اسم الملف الجديد، ومكان <extension> لاحقة الملف.



**نظام التشغيل لينكس**

**🔹 تعريف نظام لينكس**

**لينكس (Linux)** هو **نظام تشغيل حر ومفتوح المصدر** يشبه أنظمة UNIX.  
تم تطويره أولًا عام 1991 بواسطة **لينوس تورفالدز**، ويُستخدم في الحواسيب، والخوادم، والهواتف، والأجهزة الذكية.

كلمة **UNIX (يُنِكس)** 👇

هي اسم **عائلة أنظمة تشغيل** قديمة وقوية طُوِّرت في السبعينات في مختبرات Bell Labs (من شركة AT&T).

**🔹 تعريف UNIX**

**UNIX** هو نظام تشغيل متعدد المستخدمين ومتعدد المهام، صُمِّم ليكون مستقرًا وآمنًا وفعّالًا في تشغيل الخوادم وأجهزة الشبكات.

**🔹 العلاقة بين UNIX و Linux**

* **لينكس (Linux)** ليس نسخة من UNIX، لكنه **مبني على فكرته ومبادئه**.
* أي أن لينكس **يعمل مثل UNIX** في طريقة إدارة الملفات، والمستخدمين، والعمليات، لكنه **مفتوح المصدر ومجاني**، بينما UNIX في الأصل نظام **مغلق ومملوك** لشركات مختلفة.

**🔹 أهم أنظمة تنتمي لعائلة UNIX**

* **macOS** من شركة Apple
* **AIX** من IBM
* **Solaris** من Oracle
* **HP-UX** من Hewlett-Packard

**🔹 معنى "مفتوح المصدر"**

كلمة **مفتوح المصدر (Open Source)** تعني أن **شفرة النظام متاحة للجميع**،  
أي يمكن لأي شخص الاطلاع عليها، تعديلها، وتوزيعها بحرية — بشرط الالتزام برخصته (مثل GPL).

هذا يجعل لينكس نظامًا مرنًا، شفافًا، وسهل التطوير من قبل المجتمع.

**🔹 مميزات نظام لينكس**

1. **مجاني ومفتوح المصدر** — لا يحتاج ترخيص مدفوع.
2. **أمان عالٍ جدًا** بفضل صلاحيات المستخدم ونظام الملفات.
3. **ثبات واستقرار** حتى في التشغيل الطويل والمستمر.
4. **خفيف وسريع الأداء** على مختلف أنواع الأجهزة.
5. **قابل للتخصيص بالكامل** من النواة إلى الواجهة الرسومية.
6. **مجتمع ضخم ودعم واسع** من المطورين حول العالم.
7. **يدعم البرمجة والشبكات والخوادم** بشكل ممتاز.

**🔹 لماذا يُعتبر لينكس الأفضل من ناحية الأمان**

* يعتمد على **نظام صلاحيات المستخدمين**: المستخدم العادي لا يستطيع تعديل النظام بدون صلاحيات “root”.
* البرامج تعمل في **بيئة معزولة**، مما يقلل خطر الفيروسات.
* **التحديثات الأمنية سريعة ومفتوحة** للجميع.
* نادر جدًا أن تُصيب الفيروسات نظام لينكس مقارنة بأنظمة أخرى.

لذلك يُستخدم لينكس في أغلب **الخوادم، مراكز البيانات، وأجهزة الأمن السيبراني**.

**🔹 يعمل على بيئتين (Two Modes)**

لينكس يعمل على بيئتين رئيسيتين:

1. **بيئة المستخدم (User Space)** — حيث تعمل البرامج العادية والمكتبات.
2. **بيئة النواة (Kernel Space)** — حيث يعمل نواة النظام وتتعامل مباشرة مع العتاد.

هذا الفصل يحمي النظام من الأعطال ويحافظ على الاستقرار والأمان.

**🔹 أهم توزيعات لينكس (Distributions)**

**التوزيعة (Distribution)** هي نسخة من لينكس تحتوي على النواة + أدوات + واجهة + برامج جاهزة.

أشهر التوزيعات:

1. **Ubuntu (أوبونتو)**
   * سهلة الاستخدام، مناسبة للمبتدئين والمكاتب، بدعم قوي من شركة Canonical.
2. **Debian (ديبيان)**
   * نظام مستقر جدًا ومصدر لمعظم التوزيعات الأخرى (مثل أوبونتو).
3. **Fedora (فيدورا)**
   * مدعومة من Red Hat، تركّز على آخر التقنيات والتحديثات السريعة.
4. **Red Hat Enterprise Linux (RHEL)**
   * موجه للشركات والخوادم، بنظام دعم رسمي مدفوع.
5. **Kali Linux (كالي لينكس)**
   * مخصصة لاختبار الاختراق والأمن السيبراني، تضم أدوات حماية وتحليل قوية.
6. **Arch Linux (آرتش)**
   * للمستخدمين المتقدمين، تمنح تحكمًا كاملاً في إعداد النظام من الصفر.

**🧠 تخيّل المشهد:**

أنت تكتب أمرًا في لينكس، مثل:

ls

(يعني: أعرض قائمة الملفات في المجلد الحالي)

**🧩 الخطوات بالتسلسل:**

**1. 👤 المستخدم (User)**

* هو الشخص الذي يتفاعل مع النظام.
* يعطي أوامر عبر لوحة المفاتيح أو الماوس.
* مثلاً: يكتب أمرًا في الطرفية (Terminal).

**2. 💬 الصدفة (Shell)**

* الصدفة أو **Shell** هي **المترجم بين المستخدم والنظام**.
* تستقبل الأوامر التي تكتبها (مثل ls, cd, mkdir) وتحوّلها إلى تعليمات يفهمها النظام.
* أشهر أنواع الصدفة في لينكس هي: **bash**, **zsh**, **sh**.

مثال: عندما تكتب ls  
الصدفة تعرف أن هذا يعني “نفّذ برنامج عرض الملفات”.

**3. 📚 مكتبات النظام (System Libraries)**

* هذه المكتبات هي **الوسيط البرمجي** بين البرامج والنواة.
* مثل مكتبة **glibc** التي تحتوي على أوامر جاهزة للوصول إلى وظائف النظام.
* بدل ما البرنامج يتعامل مباشرة مع النواة، يستخدم هذه المكتبات لتسهيل العملية.

الصدفة (أو أي برنامج) تستدعي دوال من المكتبات لتنفيذ المهام — مثل قراءة ملف أو طباعة نص.

**4. ⚙️ النواة (Kernel)**

* هي **قلب نظام لينكس**.
* تتعامل مباشرة مع العتاد (Hardware) وتنفّذ الطلبات القادمة من المكتبات.
* مسؤولة عن:
  + إدارة الذاكرة (RAM)
  + تشغيل العمليات (Processes)
  + التعامل مع الملفات
  + التواصل مع الأجهزة عبر السائقين (Drivers)

النواة تستقبل الطلب مثلاً: “اقرأ هذه الملفات من القرص” وتنفذه.

**5. 💽 العتاد (Hardware)**

* هو الجزء المادي من الحاسوب:  
  **المعالج، الذاكرة، القرص، الشاشة، لوحة المفاتيح، إلخ.**
* النواة تتحدث معه عبر **السائقين (Drivers)**، وهي برامج صغيرة خاصة بكل قطعة.
* بعد تنفيذ العملية، تعود النتيجة للنواة → للمكتبات → للصدفة → إلى المستخدم.

**🔁 مثال مبسّط لتوضيح التدفق:**

المستخدم يكتب: cat file.txt  
⬇️  
الصدفة تترجم الأمر وتستدعي البرنامج المناسب  
⬇️  
البرنامج يستخدم مكتبات النظام لطلب “اقرأ الملف”  
⬇️  
النواة ترسل طلب القراءة إلى سائق القرص  
⬇️  
العتاد (القرص) يقرأ البيانات ويعيدها للنواة  
⬇️  
النواة → المكتبات → الصدفة → تعرض لك محتوى الملف على الشاشة

**🔹 باختصار تسلسلي:**

👤 المستخدم  
➜ 💬 الصدفة (Shell)  
➜ 📚 مكتبات النظام  
➜ ⚙️ النواة (Kernel)  
➜ 💽 العتاد (Hardware)

هل ترغب أن أعمل لك **رسم تخطيطي بسيط** يوضّح هذا التسلسل بشكل مرئي؟

**أنت قلت:**

ممكن تضيف في الشرح لماذا توجد مكتبات النظام والصدفة ليش مايكون مثل ويندوز مستخدم ثم البرامج ثم النظام ثم العتاد

**قال ChatGPT:**

أكيد، خلينا نبسط الموضوع مع توضيح **السبب وراء وجود الصدفة (Shell) ومكتبات النظام في لينكس** ولماذا لا يكون الأمر مثل ويندوز مباشرة:

**🧩 مقارنة بين لينكس وويندوز**

**1️⃣ ويندوز (تسلسل بسيط)**

* المستخدم → البرنامج → نظام التشغيل → العتاد
* في ويندوز، **معظم البرامج تأتي مع واجهات رسومية جاهزة** (GUI) تدير كل شيء تقريبًا بدون تدخل المستخدم في سطر الأوامر.
* النظام يحتوي على مكتبات أساسية، لكن البرامج غالبًا **لا تعتمد على الصدفة**؛ كل شيء مدعوم مباشرة بالواجهة الرسومية.

**2️⃣ لينكس (تسلسل مع الصدفة والمكتبات)**

* المستخدم → الصدفة (Shell) → مكتبات النظام → النواة → العتاد
* **لماذا الصدفة؟**
  + لينكس يُركز على **مرونة التحكم والبرمجة**.
  + الصدفة تتيح للمستخدم تنفيذ أوامر صغيرة، دمجها، كتابة سكريبتات، وأتمتة العمليات.
  + بدون الصدفة، لن يستطيع المستخدم التحكم الكامل في النظام بسهولة.
* **لماذا مكتبات النظام؟**
  + لتسهيل الاتصال بين البرامج والنواة.
  + كل برنامج لا يحتاج أن يعرف كيف يتعامل مع الهاردوير مباشرة.
  + مثال: printf في البرنامج يستخدم مكتبة glibc، والمكتبة تتحوّل إلى استدعاء للنواة، والنواة تتعامل مع الشاشة أو الطابعة.
  + هذا يقلل أخطاء البرمجة ويزيد الاستقرار.

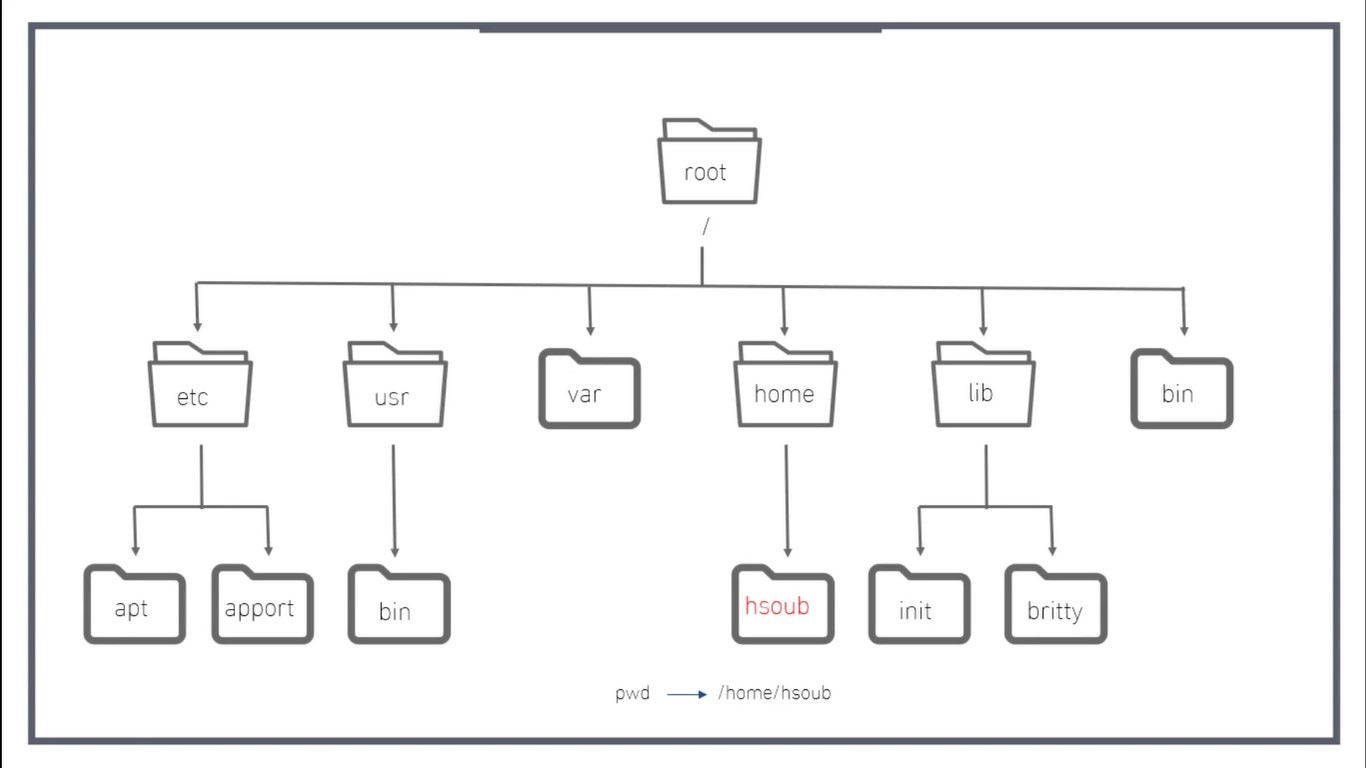
**🔹 الخلاصة**

* في **ويندوز**: معظم البرامج تأتي جاهزة مع واجهة رسومية → لا يحتاج المستخدم لمعرفة الصدفة أو المكتبات.
* في **لينكس**: الصدفة + المكتبات تمنح **مرونة كبيرة، أمان أعلى، وامكانية أتمتة** كل شيء.
* لذلك التسلسل يختلف:

المستخدم → الصدفة → مكتبات → النواة → العتاد

* هذا هو السبب الذي يجعل لينكس قوي جدًا للمبرمجين، الخوادم، والأتمتة، بينما ويندوز أسهل للمستخدم العادي مع الواجهة الرسومية.

**شجرة الملفات في لينكس**



أوامر في الترمنال في لينكس

لعرض مسار الملف الحالي:

pwd

للعودة إلى المجلد الأب نستخدم نقطتين:

cd ..

للذهاب إلى المجلد bin :

cd usr/bin

للذهاب للمسار المنزل نستخدم إشارة التيلدا tilde

cd ~

لعرض الملفات ضمن مجلد آخر مختلف في المسار عن الملف الحالي:

ls /<folder\_name>

نضيف مكان <folder\_name> مسار المجلد الذي نرغب بعرض ملفاته.

لعرض معلومات تفصيلية أكثر عن الملفات الموجودة ضمن المسار الحالي:

ls -l

لعرض جميع الملفات بالإضافة إلى الملفات والمجلدات المخفية ضمن المسار المطلوب:

ls -a

يمكن دمج الأوامر أيضا، فلعرض الملفات والمجلدات المخفية بالتفصيل نستخدم الأمر:

ls -la

إظهار تفاصيل موجزة عن محتويات ملف معين:

file <file\_name>

نضيف مكان <file\_name> اسم الملف.

لإظهار محتويات ملف نصي:

less /<file\_directory>

نضيف مكان <file\_directory> مسار الملف النصي.

الخروج من الملف النصي نضغط على حرف `q` في لوحة المفاتيح.

لمعلومات أكثر عن سطر الأوامر وبالتحديد أمر ls يمكنك قراءة مقالة [احترف الأمر ls في لينكس](https://academy.hsoub.com/devops/linux/%D8%A7%D8%AD%D8%AA%D8%B1%D9%81-%D8%A7%D9%84%D8%A3%D9%85%D8%B1-ls-%D9%81%D9%8A-%D9%84%D9%8A%D9%86%D9%83%D8%B3-r558/).

* إظهار جميع الملفات التي تبدأ بالأحرف التي نحددها
* <characters>\*

نضيف مكان <characters> الأحرف التي تبدأ بها أسماء الملفات التي نبحث عنها.

* إظهار أي ملف يبدأ بالكلمة image ويتبعها محرف واحد:
* image?
* عرض أي ملف يبدأ بأحد الحروف الموجودة بين قوسين []:
* [<characters>]\*

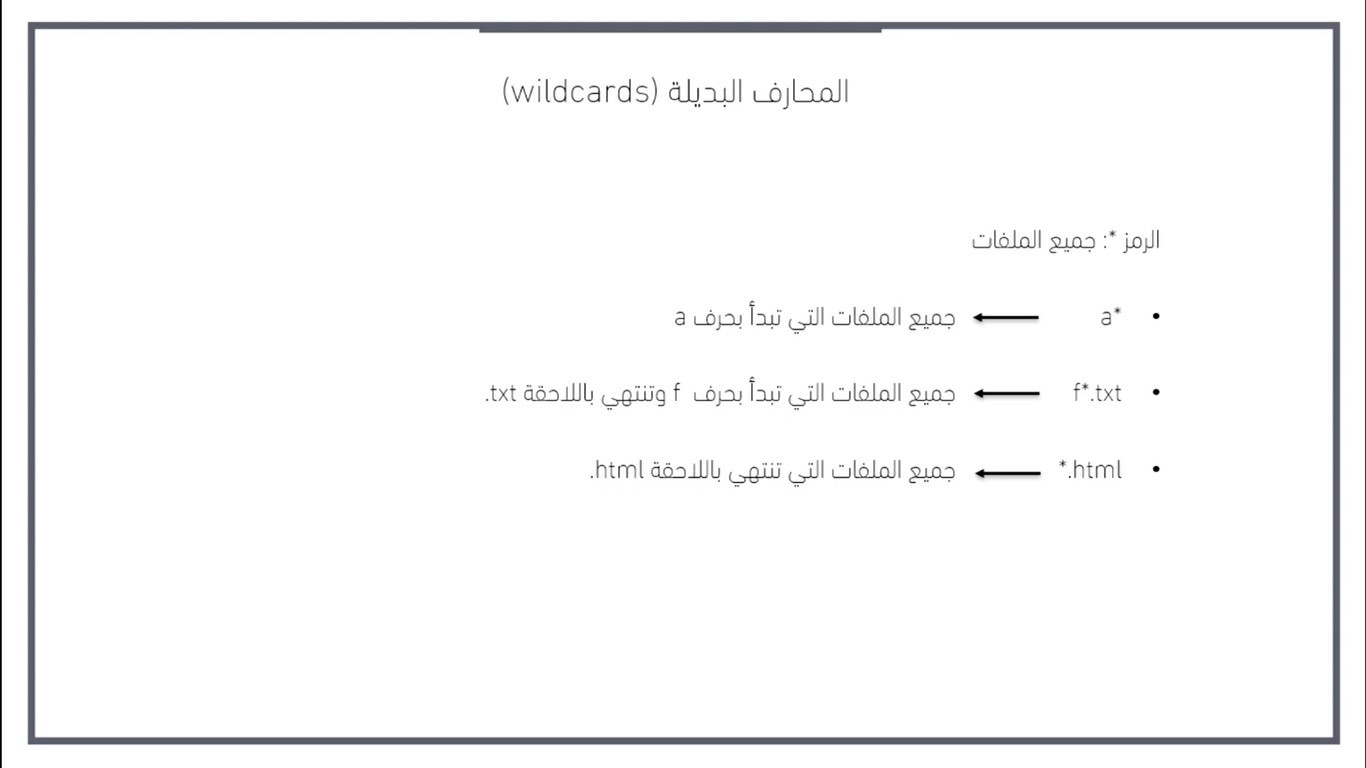
نضيف مكان <characters> الأحرف التي تبدأ بواحد منها أسماء الملفات التي نبحث عنها. ويمكن أيضًا استخدام مجال أرقام [9-0]

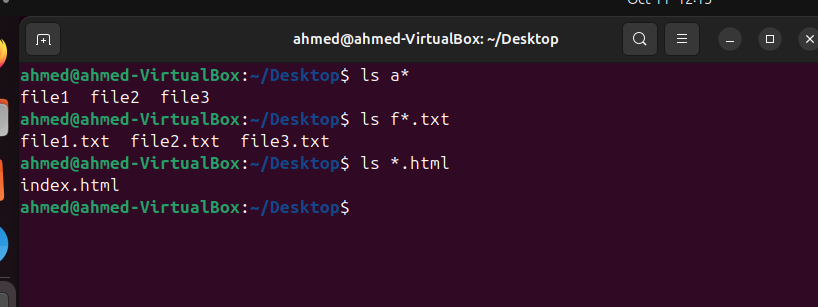
* عرض أي ملف لا يبدأ بأحد الحروف التي نقوم بتبيانها:
* [!abc]\*

يعني هذا الأمر عرض أي ملف لا يبدأ بالحروف a و b و c

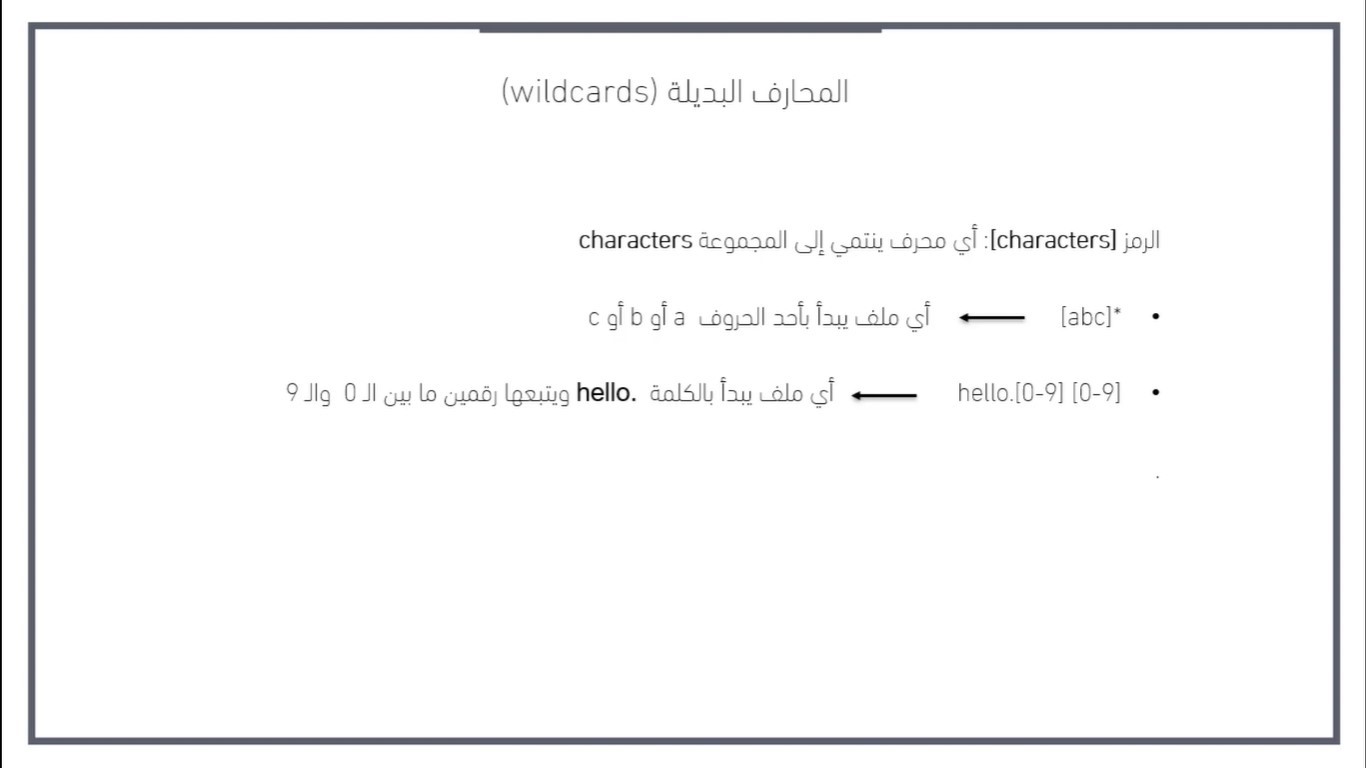
* عرض الملفات التي تبدأ بكلمة pass :
* ls etc/pass\*
* عرض جميع الملفات التي تنتهي باللاحقة conf :
* ls /etc/\*.conf

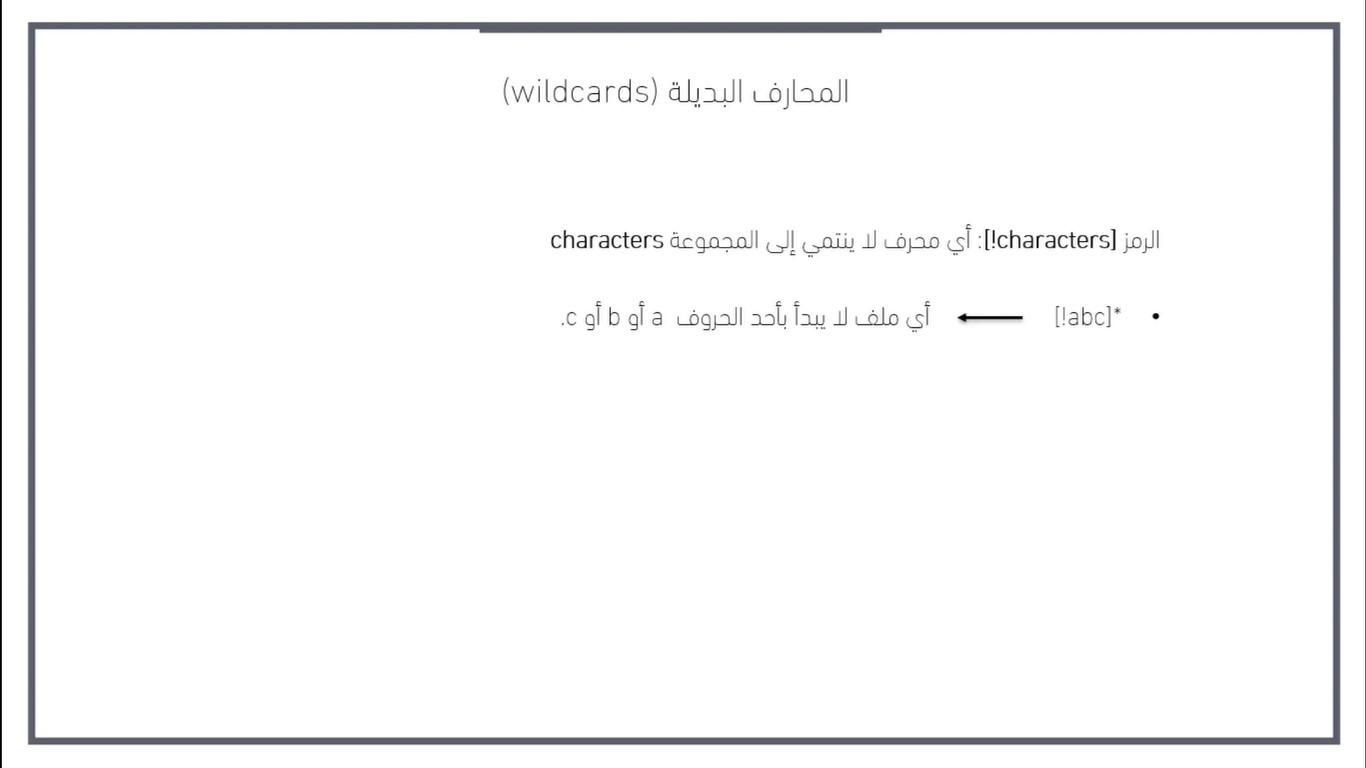
**المحارف البديلة**











**نسخ الملفات وانشاء المجلدات**

تغيير مسار الملف عبر طريقة المسارات المطلقة:

cd /home/hsoub/Desktop

إنشاء مجلد أب بإسم a وبداخله مجلد b بداخل المجلد b إنشاء المجلد c

mkdir -p a/b/c

إنشاء مجلد مخفي بإسم test:

* mkdir .test

نسخ ملف إلى مجلد جديد:

cp <file\_name> <directory>

نضيف مكان <file\_name> اسم الملف المراد نسخه ومكان <directory> مسار الملف الجديد. ويمكن  إضافة الخيارات:

* `a-` نسخ الملفات والمجلدات مع كل خاصياتهم.
* ` i-` طلب موافقة المستخدم قبل استبدال ملف موجود مسبقا.
* `v-` لإظهار معلومات حول عملية النسخ.

:

* cp -r <folder\_name> <folder\_directory>
* نضيف مكان <folder\_name> اسم المجلد المراد نسخه ومكان <folder\_directory> مسار المجلد الجديد.
* يوفر الخيار r- نسخ المجلدات بجميع محتوياتها.

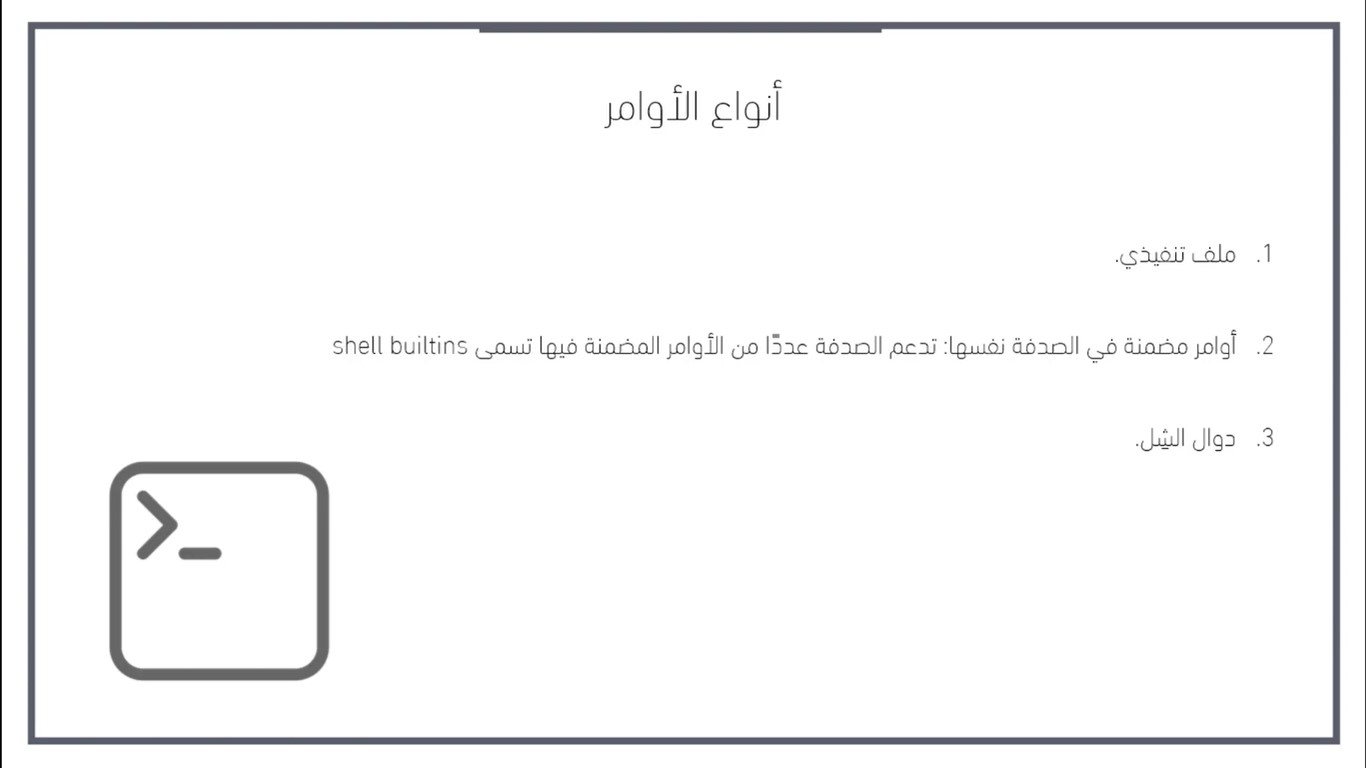


**نقل الملفات والمجلدات**



**الأوامر المساعدة(info و type و which و help و man)**

الأوامر السابقة التي درسنها التي تكتب في التيرمنال تتدرج تحت الأنواع الأتية:-



نضيف مكان <command> الأمر.

* عرض مسار الملف التنفيذي لأمر ما:

which <command>

* عرض الأوامر المضمنة في الصدفة:

help <command>

* عرض معلومات تفصيلية عن أمر معين:

<command> --help

* تنظيف محاكي الطرفية:

clear

* مشاهدة صفحة الدليل لأمر معين:

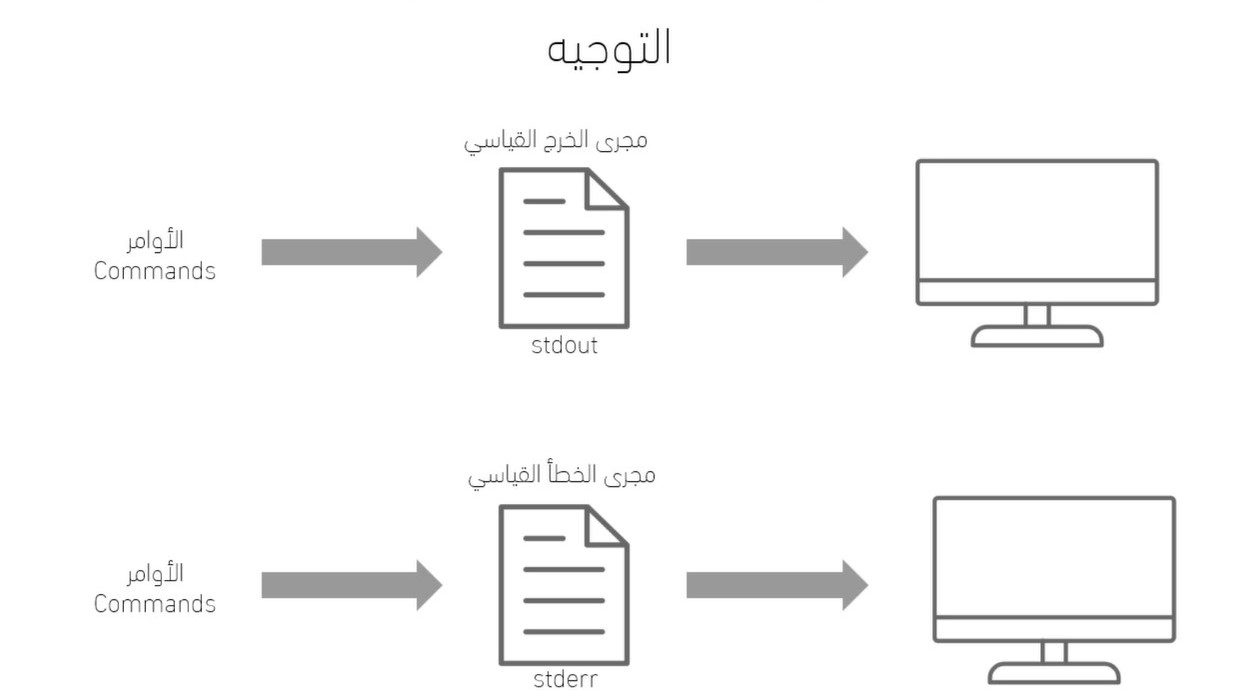
man <command>

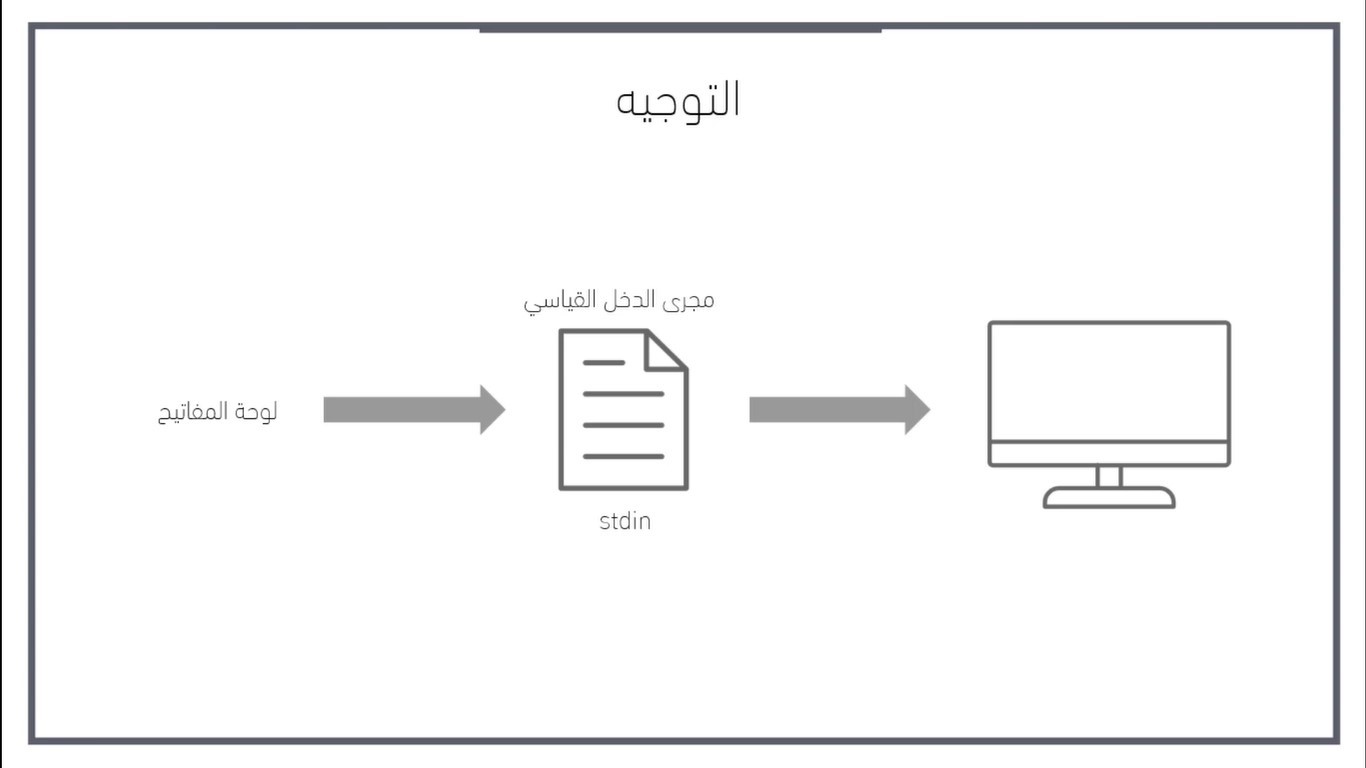
* مشاهدة صفحة المعلومات info لأمر معين:

**محرر النصوص nano**



**مدخل لإعادة التوجيه**





**كل شي في لينكس ملف**

في نظام **لينكس** توجد قاعدة أساسية:

**كل شيء هو ملف**

🔹 **معنى ذلك:**  
لينكس يتعامل مع كل شيء كأنه ملف، سواء كان:

* ملف نصي
* جهاز (قرص صلب، لوحة مفاتيح، شاشة)
* أو منافذ الإدخال والإخراج

**ملفات الادخال والإخراج القياسية:**

* **stdin (0)** → *ملف الإدخال القياسي*، عادةً من لوحة المفاتيح.
* **stdout (1)** → *ملف الخرج القياسي*، عادةً إلى الشاشة.
* **stderr (2)** → *ملف الخطأ القياسي*، عادةً أيضًا إلى الشاشة.

كل ملف قياسي له رقم معرف:

* 0 = stdin
* 1 = stdout
* 2 = stderr

**توجيه الملفات**

يمكنك تحويل أو إعادة توجيه هذه الملفات بسهولة:

* قراءة من ملف بدل لوحة المفاتيح:

command < input.txt

* إرسال الخرج إلى ملف بدل الشاشة:

command > output.txt

* حفظ الأخطاء في ملف آخر:

command 2> errors.txt





**تعليق توزيعة ابونتو**

ام ابونتو علق عند تحميله فتحت سطر الاوامر ابش اكتب

**شات جي بي تي**

تمام 👍  
خلنا نحاول نساعدك خطوة بخطوة.  
لكن أولاً أحتاج أعرف بعض التفاصيل:

1. هل النظام يعلق **أثناء الإقلاع (boot)** قبل ما تدخل سطح المكتب؟
2. أم يعلق بعد الدخول لسطح المكتب أثناء التشغيل؟
3. وهل أنت الآن في **سطر الأوامر (tty)** (يعني شاشة سوداء فيها login:)؟

**لوكنت في سطر الأوامر بعد ماعلق النظام**

سجّل دخولك بكتابة:

login: your\_username

Password: your\_password

بعد الدخول، جرّب هذه الخطوات حسب السبب:

**تحديث النظام واصلاح الحزم في حالة تلفها**

sudo apt update

sudo apt upgrade -y

sudo apt --fix-broken install

**تنظيف وإصلاح النظام**

sudo apt autoremove -y

sudo apt clean

**إصلاح ملفات النظام**

sudo dpkg --configure -a

sudo fsck -f /

ملاحظة: إذا طلب منك fsck تأكيد الإصلاح، اضغط **y** عدة مرات.

**محاول تشغيل واجهة الرسومية للسطح المكتب**

بعد الإصلاح، جرّب ترجع للواجهة الرسومية:

sudo systemctl restart gdm

أو إذا تستخدم واجهة أخرى:

* sudo systemctl restart lightdm
* sudo systemctl restart sddm

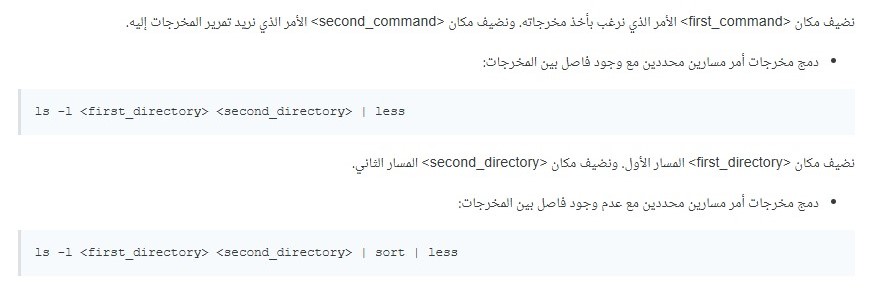
**لو ما نفع، جرّب الدخول في وضع الطوارئ**

1. أعد التشغيل.
2. عند ظهور شعار Ubuntu، اضغط **Shift** (أو **Esc** حسب نوع النظام).
3. اختر **Advanced options for Ubuntu**.
4. اختر **(recovery mode)**.
5. اختر “root” من القائمة.
6. ثم نفذ الأوامر التالية:
7. mount -o remount,rw /
8. apt update
9. apt upgrade -y

**التعرف على الأنابيب**

**pipes وعلى الأمر cat و sort**

****



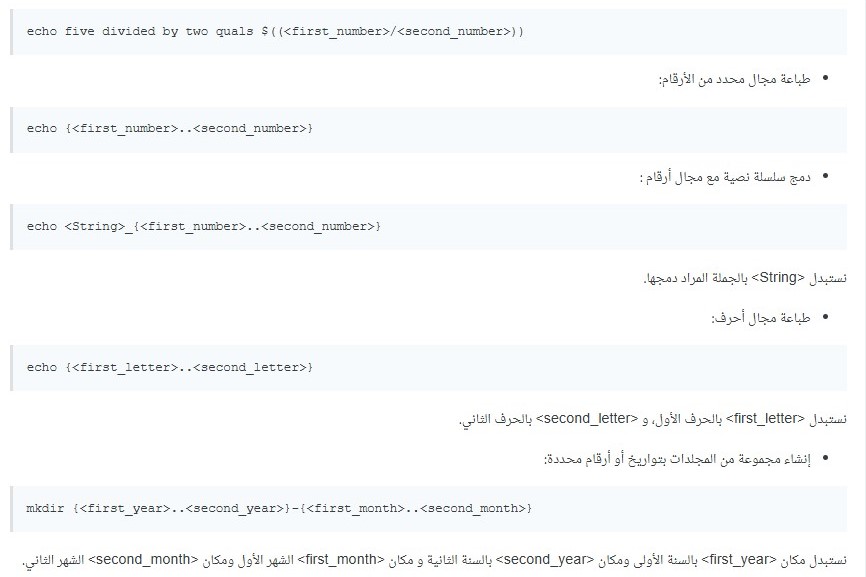
**التعرف على الأوامر wc head tail**



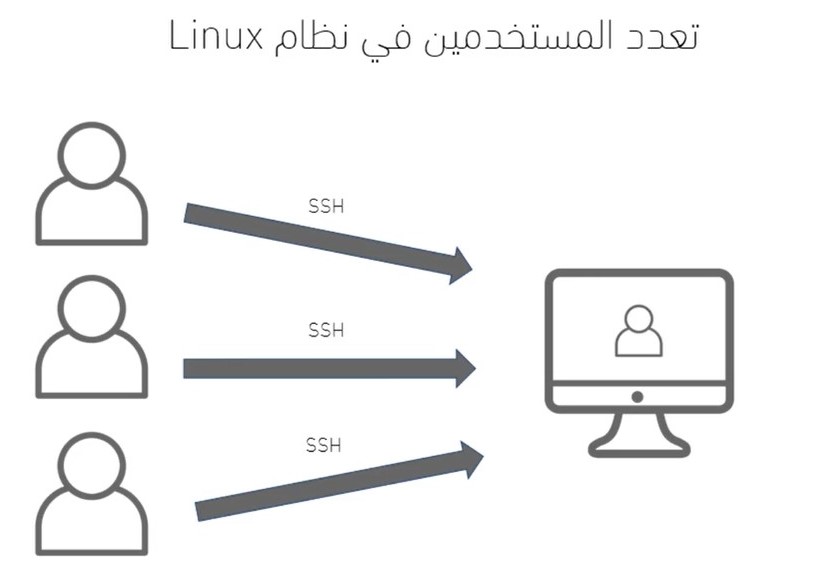


**التوسعات وجملة echo**

****



**مدخل للأذونات**

****

* إظهار معلومات حول ملف أو مجلد:

file <file\_name>

نضيف مكان <file\_name> اسم الملف أو مساره.

* إظهار معلومات حول المستخدم الحالي:

id

* عرض أذونات الملفات الموجودة ضمن المسار الحالي:

ls -l

للتعرف أكثر على أذونات الملفات على لينكس ويونكس يمكنك قراءة مقالة [مبادئ أذونات الملفات (File permissions) على لينكس](https://academy.hsoub.com/devops/linux/%D9%85%D8%A8%D8%A7%D8%AF%D8%A6-%D8%A3%D8%B0%D9%88%D9%86%D8%A7%D8%AA-%D8%A7%D9%84%D9%85%D9%84%D9%81%D8%A7%D8%AA-file-permissions-%D8%B9%D9%84%D9%89-%D9%84%D9%8A%D9%86%D9%83%D8%B3-r242/).

**يحتاج لينكس، مثل غيره من أنظمة التشغيل متعدّدة المستخدمين Multi-user system إلى طريقة للتحكّم في وصول المستخدمين إلى مختلف الملفات. ليس من المحبّذ مثلا أن يعدّل مستخدم آخر على ملفات الإعداد التي أخذ ضبطها كثيرا من وقتك. يُطبَّق مبدأُ الفصلِ هذا على نظام تشغيل لينكس بآلية الأذونات Permissions.** [](https://academy.hsoub.com/uploads/monthly_2016_03/56f5be6d484b7_linux-file-permissions(1).png.83ff47a9b66cf7fc09bd650e0ef89c67.png)

**سؤال ممتاز 👏  
في نظام لينكس (Linux)، كل ملف أو مجلد له صلاحيات (Permissions) تحدد من يُسمح له بعمل ماذا عليه.  
وهذه الصلاحيات مقسمة إلى ثلاث أنواع من الأذونات (permissions):**

**🔹 1. إذن القراءة (Read - r)**

* **يرمز له بالحرف r.**
* **لملف: يعني أنه يمكن قراءة محتواه (عرض النص أو البيانات)**
* **لمجلد: يعني أنه يمكن رؤية أسماء الملفات داخله (عرض قائمة الملفات فقط، بدون فتحها)**

**مثال:  
إذا كان عندك ملف notes.txt وعندك إذن قراءة فقط (r--)، تقدر تستخدم cat notes.txt أو less notes.txt لعرض محتواه، لكن لا تقدر تعدّل عليه أو تحذفه.**

**🔹 2. إذن الكتابة (Write - w)**

* **يرمز له بالحرف w.**
* **لملف: يعني أنه يمكن تعديل أو حذف محتواه.**
* **لمجلد: يعني أنه يمكن إضافة أو حذف أو إعادة تسمية ملفات داخل المجلد.**

**مثال:  
إذا عندك مجلد باسم projects ومعه إذن كتابة (w)، تقدر تنشئ ملف جديد فيه (touch new.txt) أو تحذف ملف (rm old.txt).**

**🔹 3. إذن التنفيذ (Execute - x)**

* **يرمز له بالحرف x.**
* **لملف: يعني أنه يمكن تشغيل الملف كبرنامج أو سكربت.**
* **لمجلد: يعني أنه يمكن الدخول إليه (cd) والتنقل داخله.**

**مثال:  
إذا عندك سكربت run.sh ولا يملك إذن تنفيذ، لن يعمل عند كتابة ./run.sh.  
لكن إذا أضفت له إذن تنفيذ (chmod +x run.sh)، سيعمل.**

**كيف تعمل الأذونات**

**يُخزَّن كلّ ملف على النظام بأذوناته الخاصّة. كيف تُمثَّل؟ من الراجح أنك صادفتها أثناء عملك على نظام لينكس:**

drwxr-xr-x

-r-w-rwrw-

-rwxr--r--

**تظهر الأذونات عند تنفيذ أوامر مثل ls -l. تخبر سلسلة المحارف Charcters بمعلومات الملف؛ يمكن تقسيم سلسلة المحارف الخاصّة بمعلومات الملف إلى أربعة أقسام:**

**القسم الأول: هو المِحرف الأول الذي يدلّ على نوع الملفّ.**

**القسم الثاني: يحوي المحارف من 2 إلى 4؛ وهي أذونات المستخدِم مالك الملف؛ أي ما يحقّ لهذا المستخدم فعله على الملف.**

**القسم الثالث: المحارف من 5 إلى 7؛ وهي أذونات المجموعة مالكة الملف.**

**القسم الرابع: المحارف من 8 إلى 10؛ وهي أذونات بقية المستخدمين (ليسوا مالك الملف ولا ينتمون للمجموعة مالكة الملف).**

**يمكن للمحرف الأول أن يكون أحد المحارف التالية:**

**- للملفات العادية.**

**d: للمجلّدات.**

**l: للوصلات الرمزية Symbolic link.**

**s: لمقابس Socket يونكس.**

**p: لأنابيب الاتّصال.**

**c: للأجهزة الطرفيّة المحرفيّة Character device file.**

**b: للأجهزة الطرفيّة الكتليّة Block device file.**

**توضّح المحارف التسعة التالية لنوع الملف الأذونات في ثلاثة أقسام من ثلاثة محارف (القسم الثاني، الثالث والرابع المذكورة أعلاه): تعرض المحارف الثلاثة الأولى أذونات القراءة، الكتابة والتنفيذ بالنسبة لمالك الملف.**

**تعرض المحارف الثلاثة الموالية نفس الأذونات ولكن بالنسبة للمجموعة مالكة الملف؛ والمحارف الثلاثة الأخيرة الأذونات بالنسبة لبقية المستخدمين.**

**يعني المحرف الأول من كل قسم من إذن القراءة (r)، المحرف الثاني إذن الكتابة (w) والمحرف الثالث إذن التنفيذ (x). يدلّ ذكر المحرف (w، r وx) على وجود الإذن أما غياب الإذن فيُشار إليه بعارضة -.**

**يعني وجود عارضة مكان إذن القراءة (أو الكتابة أو التنفيذ) أن المعني (المالك، المجموعة أو الآخرين) ليس لديه هذا الإذن.**

**بالعودة إلى المثال:**

drwxr-xr-x

**يعني هذا السطر أننا أمام مجلّد (المحرف الأول d)، لدى مالك الملف (المحارف من 2 إلى 4) جميع الأذونات (rwx)، لدى المجموعة المالكة (المحارف من 5 إلى 7) إذنا القراءة والتنفيذ ولكن ليس لديها إذن الكتابة (r-x)، ونفس الشيء بالنسبة لبقية المستخدمين.**

**تعديل الأذونات**

**يُستخدَم أمر chmod لتعديل الأذونات. توجد طريقتان لذلك: الأولى باستخدام الأحرف المذكورة أعلاه (w، r وx)، والثانية باستخدام أرقام. سنشرح الطريقة الأخيرة هنا لأنها أسرع كثيرا.**

chmod 775 ~/my/file

**نلقي نظرة على القائمة التالية لفهم معنى الأرقام:**

**4 = r**

**2 = w**

**1 = x**

**0= إذن غير موجود**

**نجمع الأرقام للحصول على الأذونات. بالنسبة لإذن rwx فيجب أن يكون 4+2+1=7؛ إذن r-x يجب أن يكون 4+0+1=5 وهكذا.**

**بما أن لدينا ثلاثة أقسام من الأذونات (المستخدِم المالك، المجموعة المالكة وبقية المستخدمين) فسنحتاج إلى ثلاثة أرقام، رقم لكلّ قسم.**

**أي أننا نجمع الأرقام المقابلة لأذونات كل قسم ونضعها جنبا إلى جنب في عدد من ثلاثة أرقام: الرّقم على اليسار لأذونات المستخدم المالك، الرقم في الوسط للمجموعة المالكة والرقم على اليمين لبقية المستخدمين.**

**نعرف، بتنفيذ المبدأ أعلاه ، أن الأمر السابق يعدّل أذونات الملف /my/file/~ لتصبح كالتالي (طبقنا الأمر على ملف عادي، وبالتالي تظهر - في معلومات الملف تليها الأذونات):**

-rwxrwxr-x

**تعديل ملكية ملف**

**يوجد أمران لتعديل ملكية الملف، الأول وهو chown يغيّر المستخدم المالك والثاني chgrp ويغيّر المجموعة المالكة.**

**تغيير المستخدم المالك (المالك الجديد للملف /my/file/~ هو user):**

chown user ~/my/file

**تغيير المجموعة المالكة (المجموعة الجديدة هي group):**

chgrp group ~/my/file

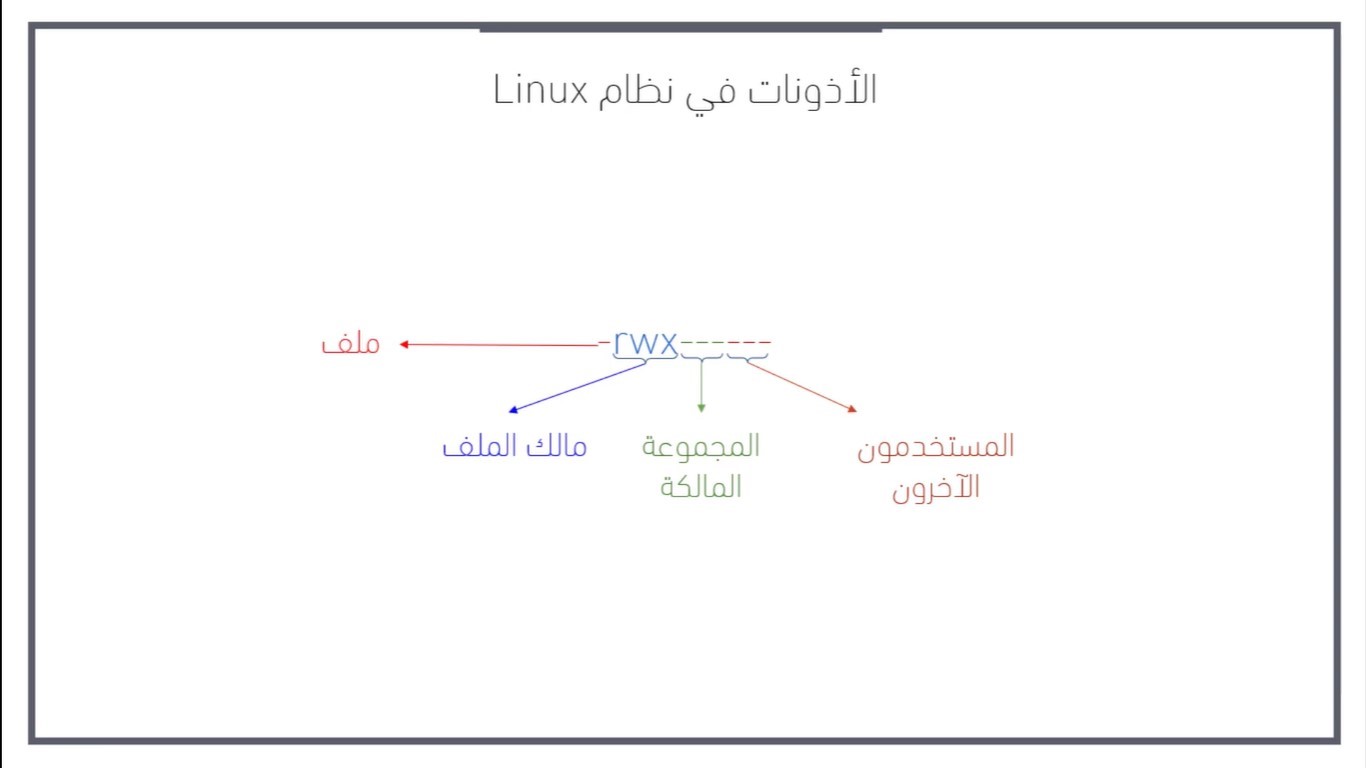
**إن لم تكن مالكَ الملف أو تنفذ الأمرين أعلاه بصلاحيات المستخدم الأعلى root فلن ينجح تنفيذ الأمر.**

ترجمة -وبتصرّف- للمقال

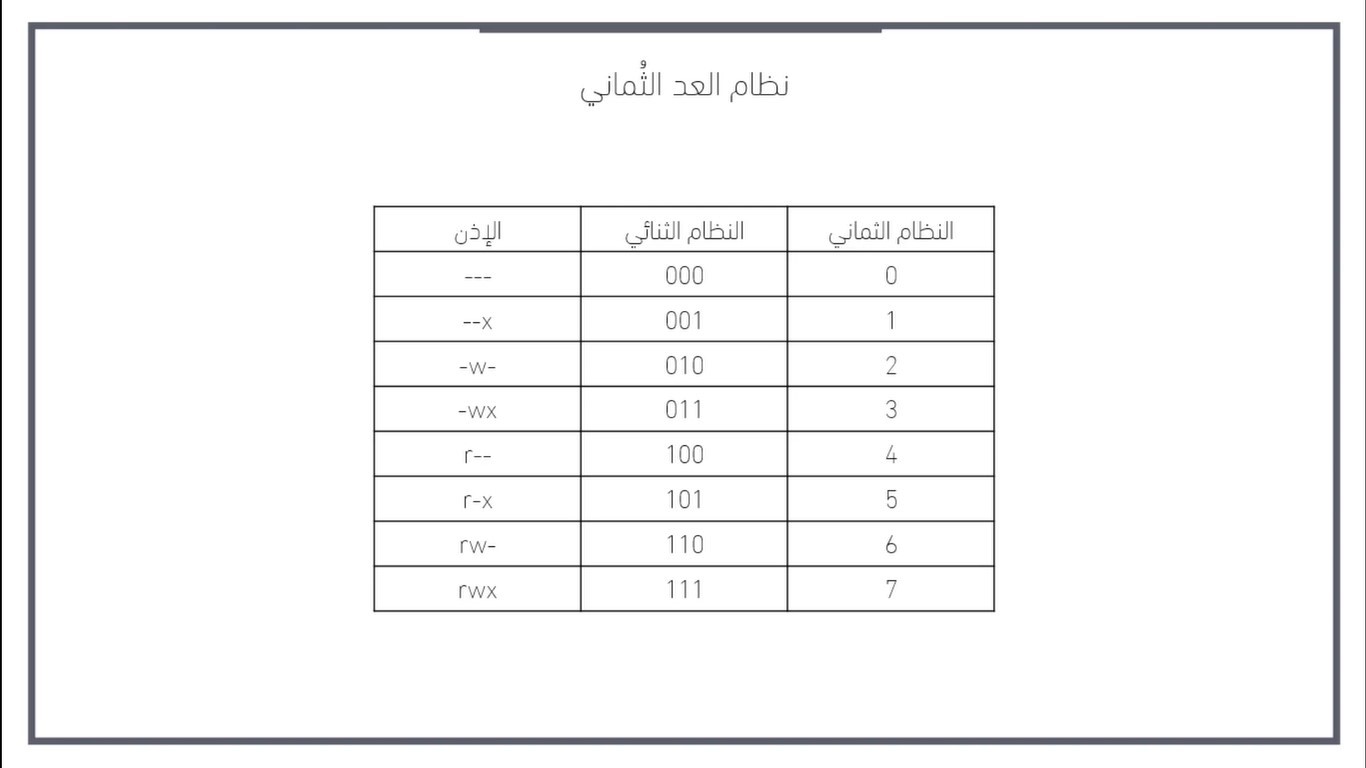
[Linux File Permissions Explained](http://radek.io/2011/07/19/linux-file-permissions-explained/)

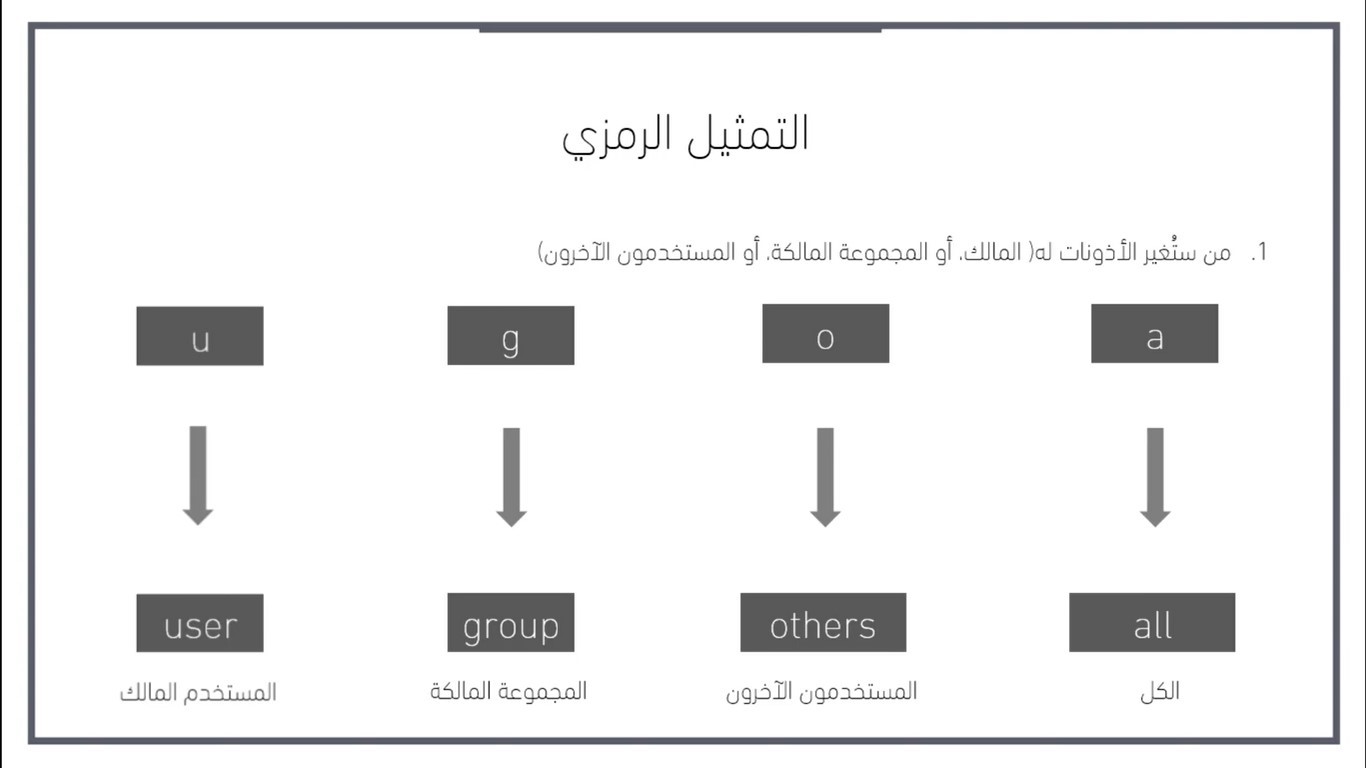
لصاحبه Radek Pazdera.



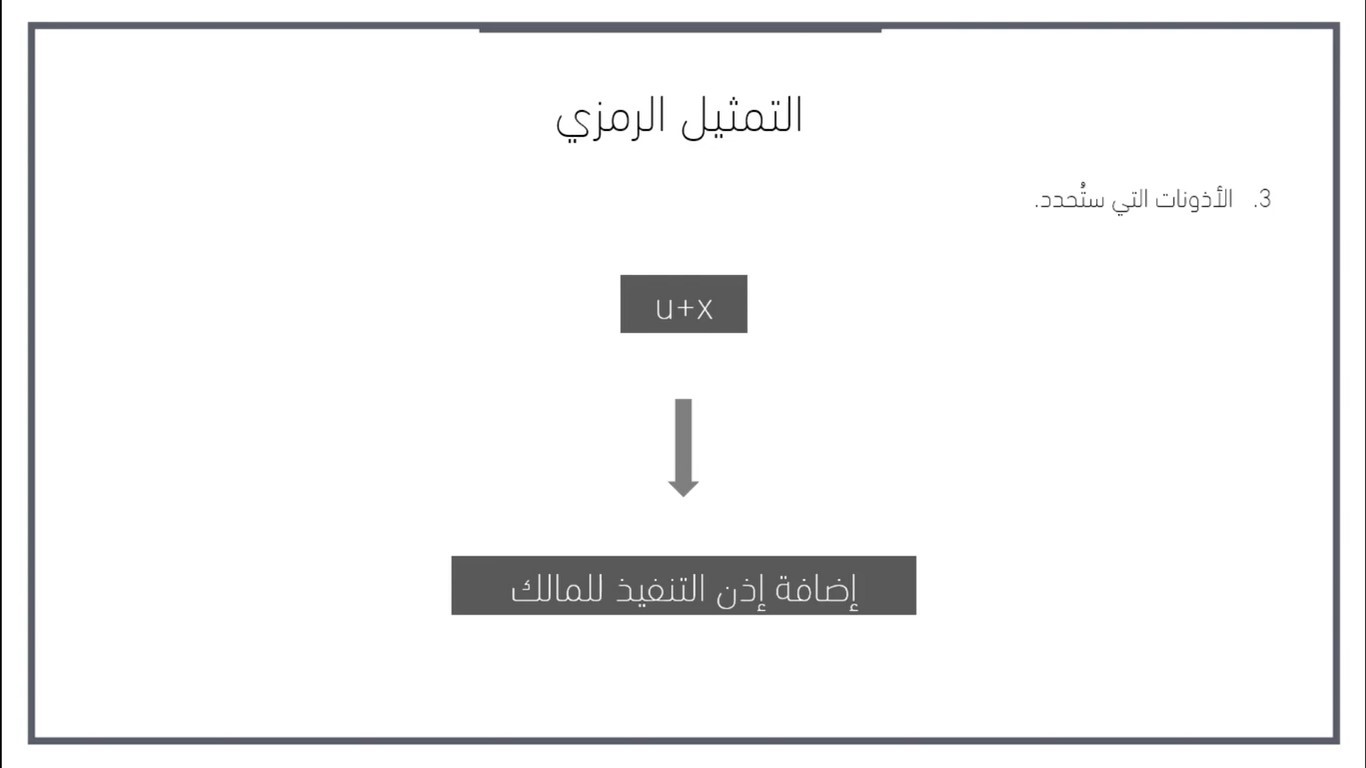


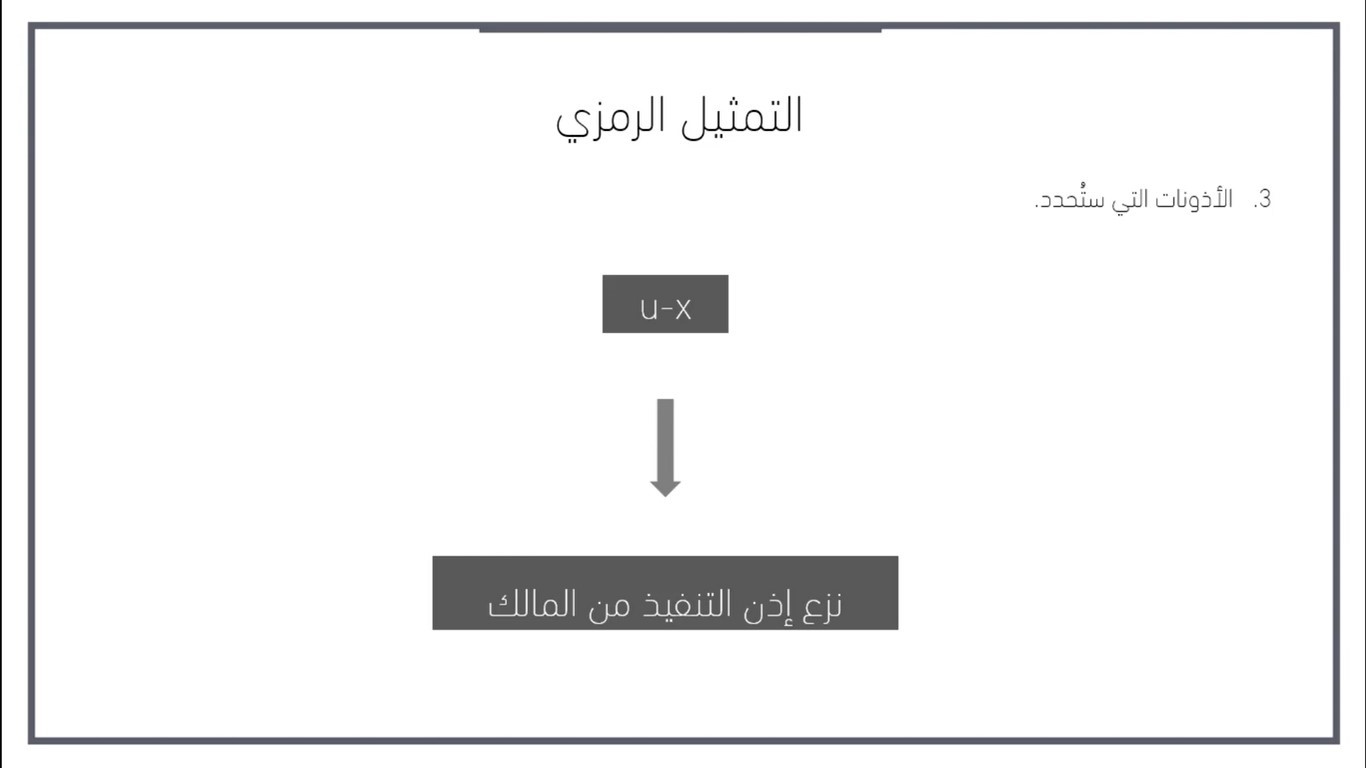
**تغيبر الأذونات** Chmod

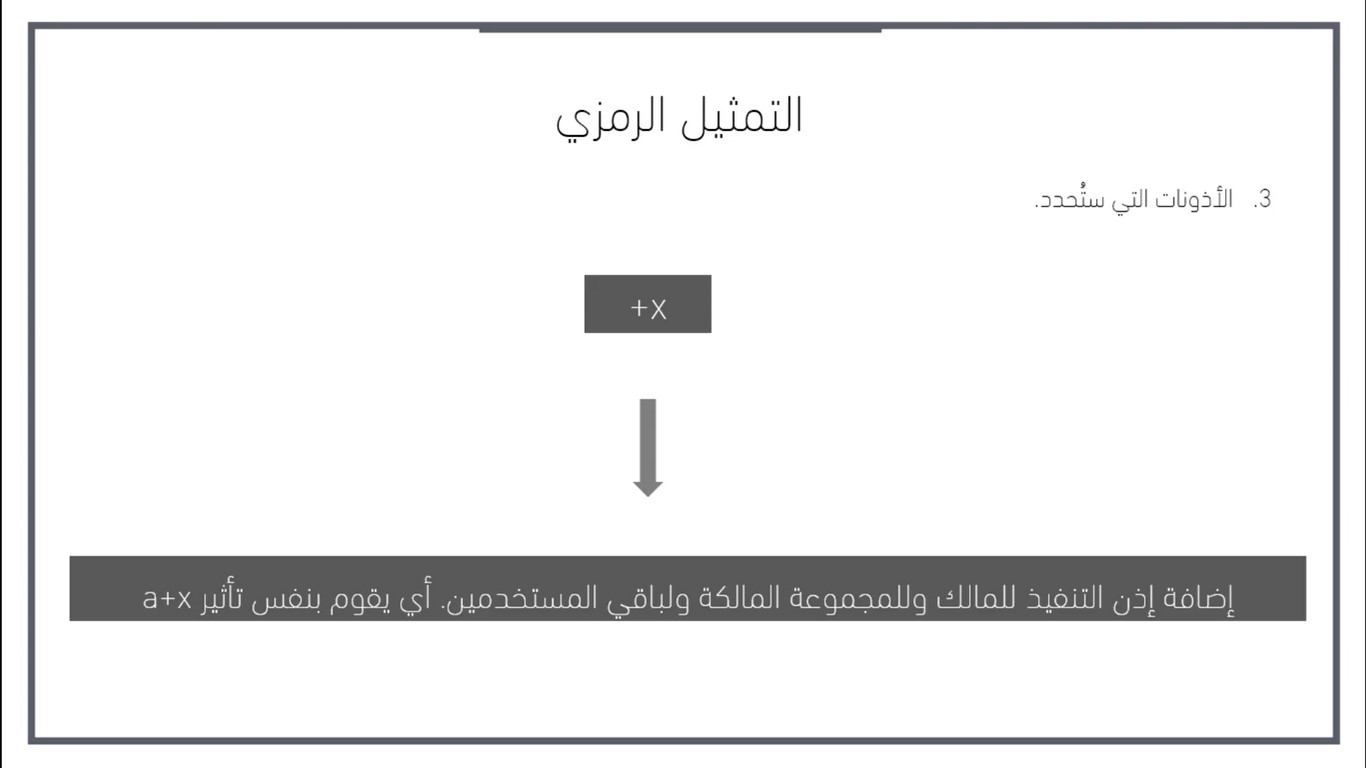


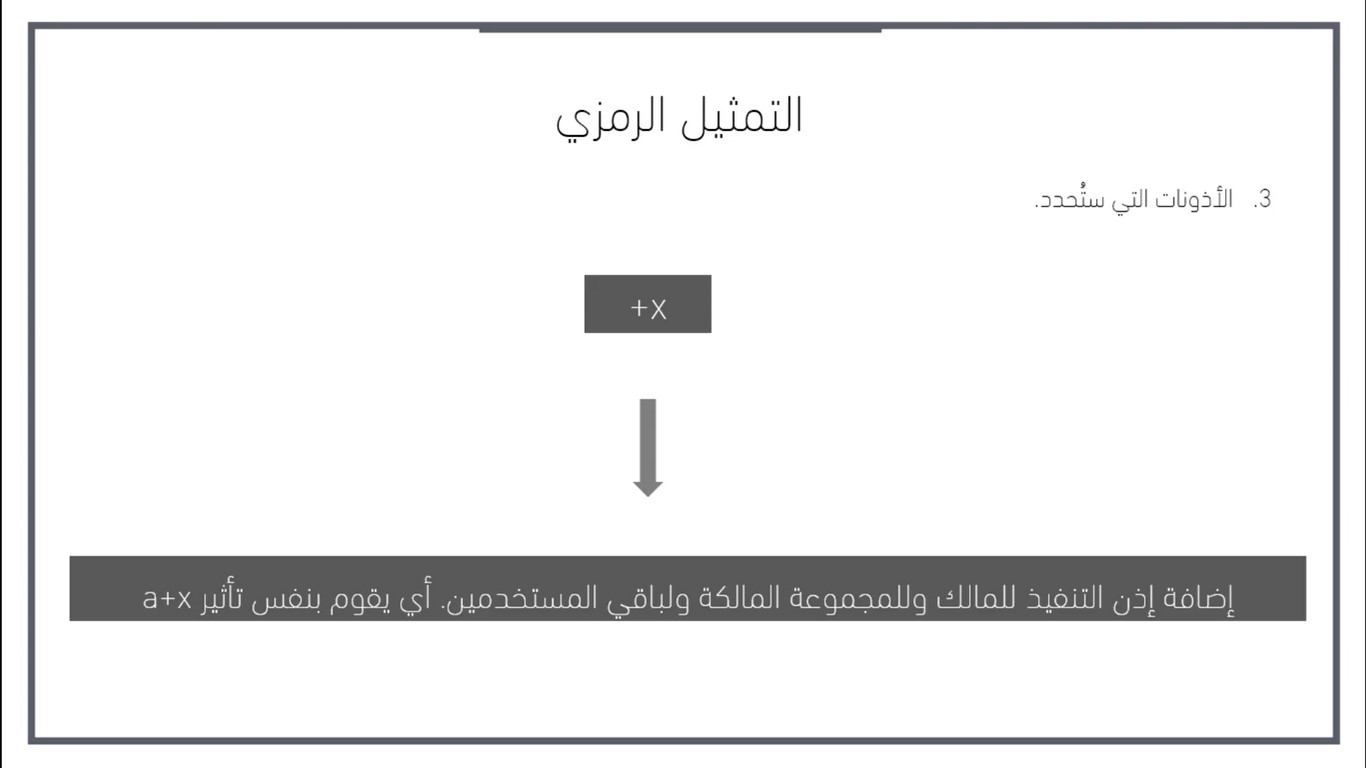
















**تنفيذ الأوامر عبر المتسخدم الجذر**

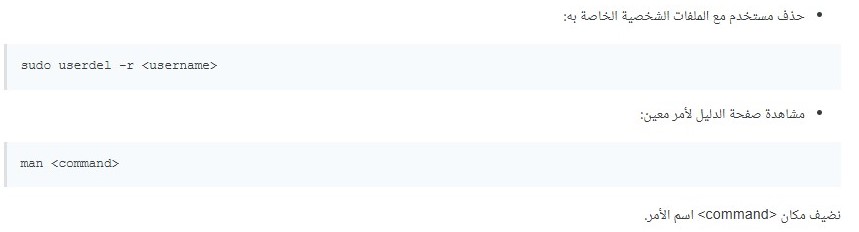




**أنواع المستخدمون في لينكس**







**الطريقة 1: عرض كل المستخدمين من ملف النظام /etc/passwd**

كل مستخدم في النظام مسجل في هذا الملف.

cat /etc/passwd

سيظهر لك شيء مثل:

root:x:0:0:root:/root:/bin/bash

user1:x:1000:1000:User One:/home/user1:/bin/bash

ftp:x:14:50:FTP User:/srv/ftp:/sbin/nologin

🔹 الجزء قبل أول نقطتين : هو **اسم المستخدم**  
مثلاً:

* المستخدم root
* المستخدم user1
* المستخدم ftp

**🧩 الطريقة 2: عرض أسماء المستخدمين فقط (بدون التفاصيل)**

إذا أردت فقط الأسماء:

cut -d: -f1 /etc/passwd

**⚙️ الطريقة 3: عرض المستخدمين البشريين فقط (الذين يمكنهم تسجيل الدخول)**

المستخدمون العاديون عادةً معرفاتهم (UID) تبدأ من 1000 فما فوق.

awk -F: '$3 >= 1000 && $3 < 65534 {print $1}' /etc/passwd

**👤 الطريقة 4: عرض المستخدمين الحاليين المسجلين في النظام الآن**

أي من قام بتسجيل الدخول حاليًا:

who

أو:

w

**🧠 ملاحظة:**

* المستخدمون **النظاميون** (مثل daemon, ftp, nobody) تُستخدم لحماية النظام وتشغيل الخدمات.
* المستخدمون **البشريون** هم الذين تضيفهم بنفسك لتسجيل الدخول.

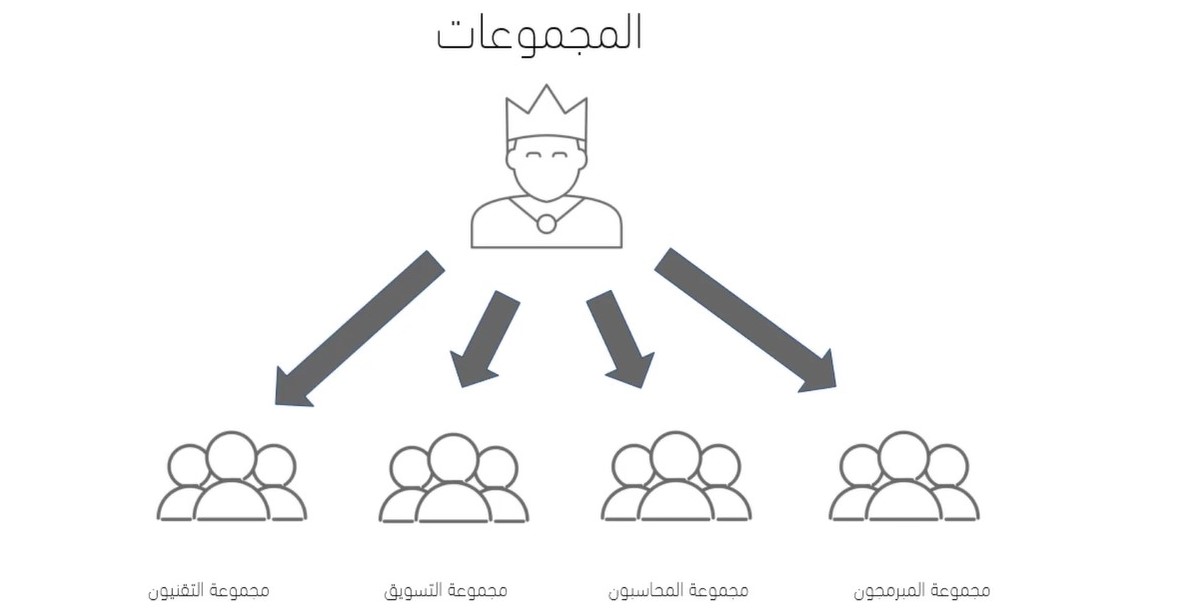
cat /etc/passwd يستخدم هذا الأمر لمعرفة أسماء المستخدمين

cat /etc/group يستخدم هذا الأمر لمعرفة أسماء المجموعات

**المجموعات** في لينكس









هذا الامر يمكننا من خلاله معرفة أسماء المتسخدمين لمجموعة معينة

sudo cat /etc/group | grep name group

تغير **المالك** والمجموعة المالكة

Chown= change ownership

اذا اضفنا للامر

Chown

للوسطاء الاتية







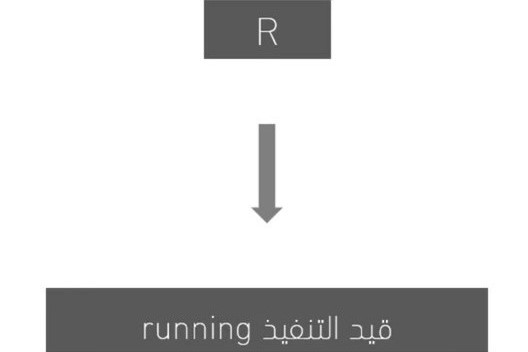




**مشاهدة** **العلميات**



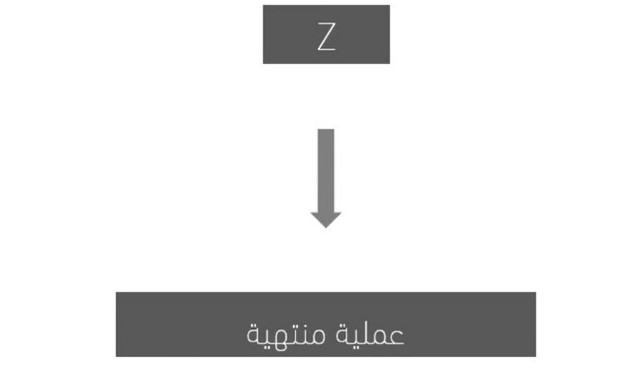
حالات العلميات في لينكس

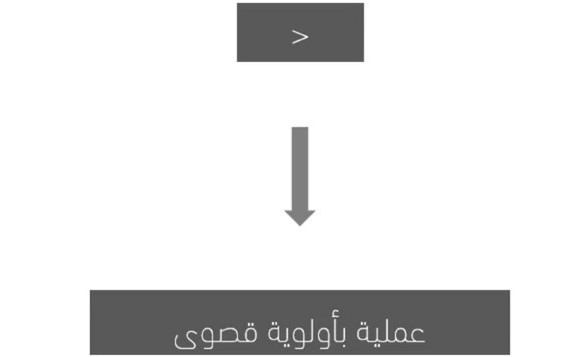














**التحكم في العمليات**

****

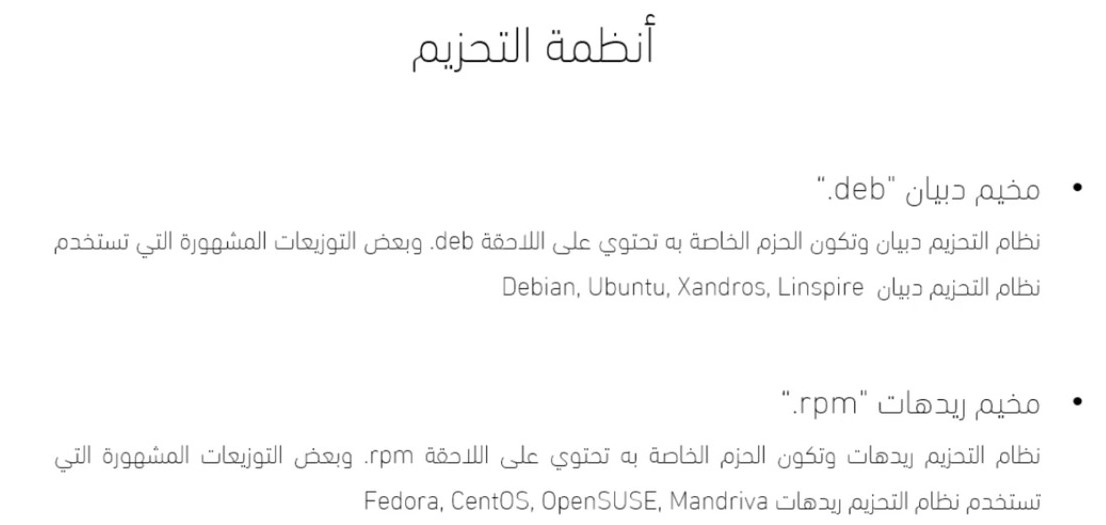
**الإشارات**



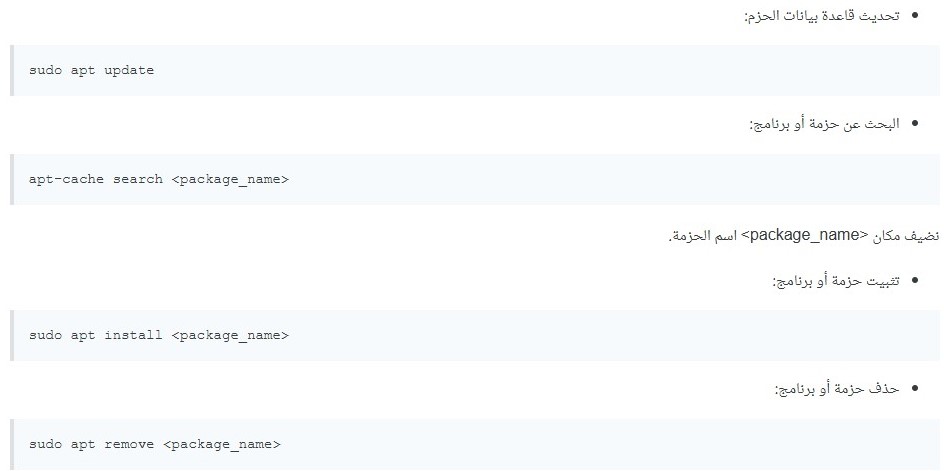


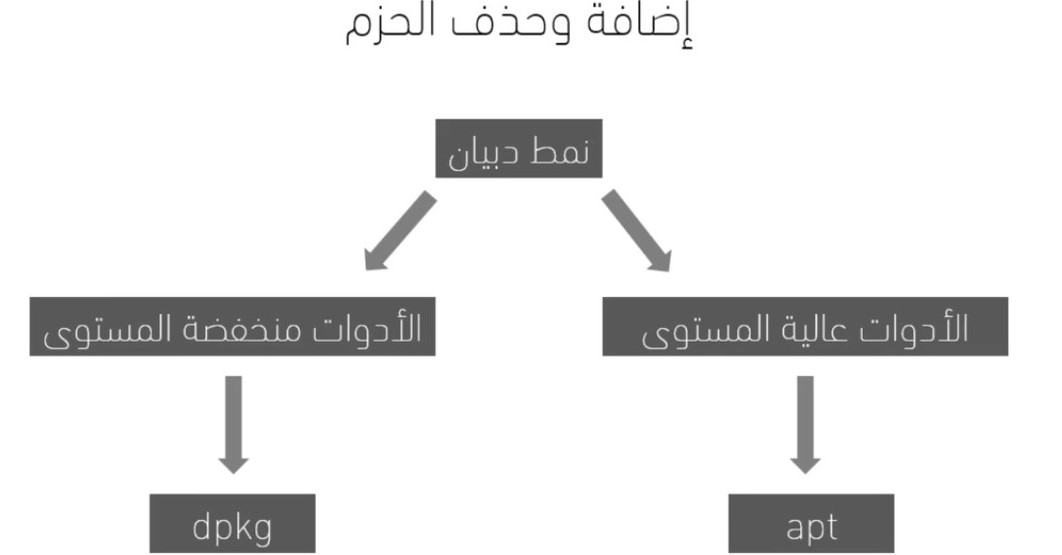


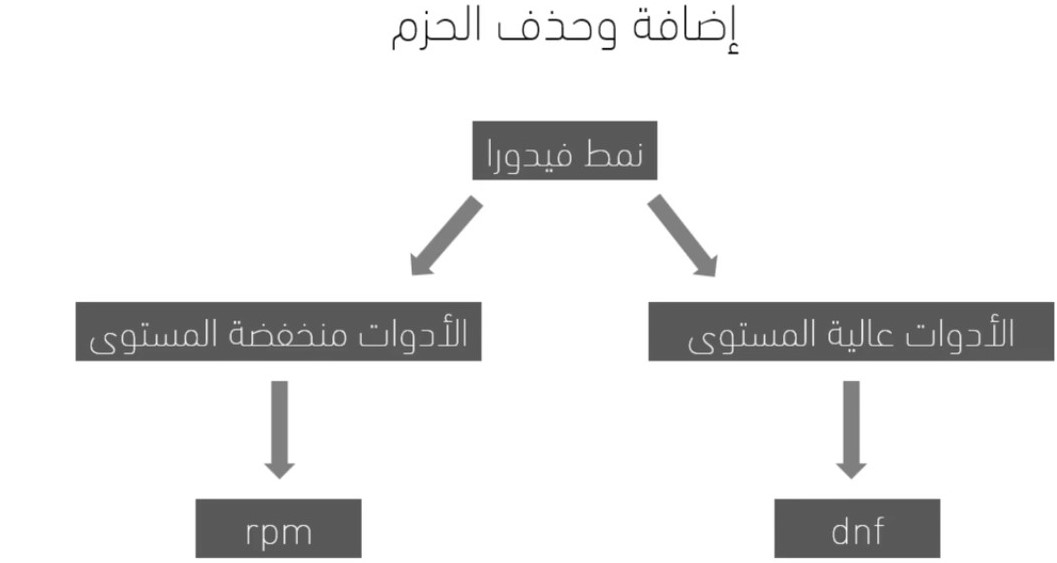
**مدخل الى إدارة الحزم**



إضافة وحذف الحزم

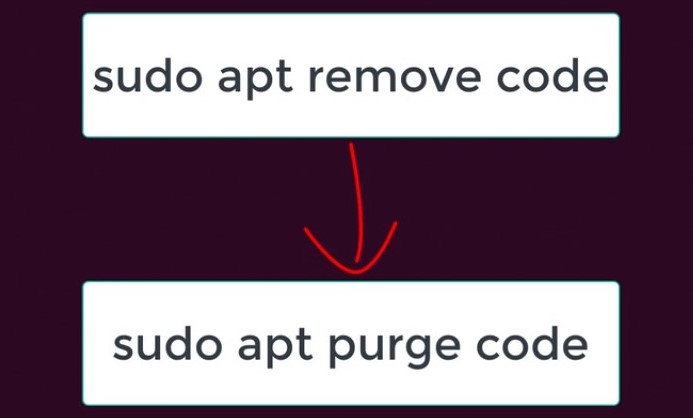




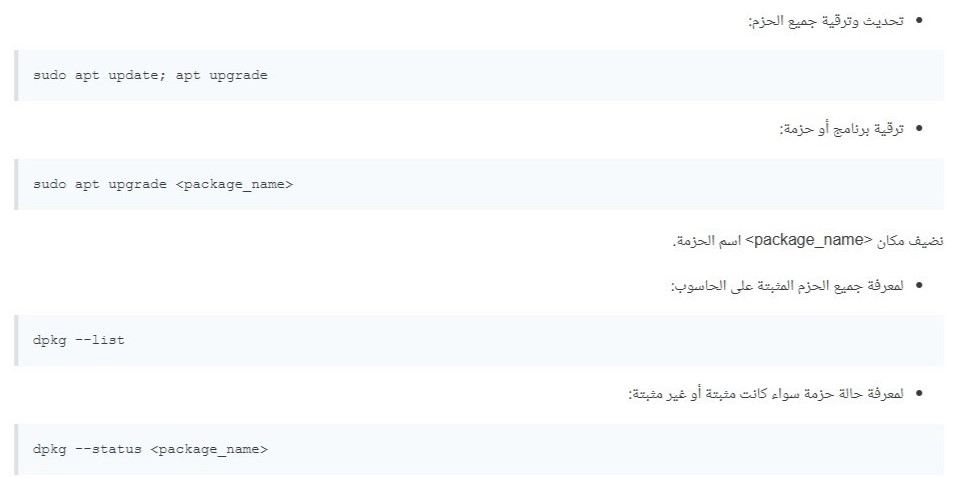


اذا اردنا حذف حزمة مع ملفات الاعداد الخاص بها أي الحذف من الجذور نستخدم الأمر

purge



ترقية وعرض الحزم المثبتة



يفضل الأمر الأول يكتب هكذا من أجل حصوله على الأذن

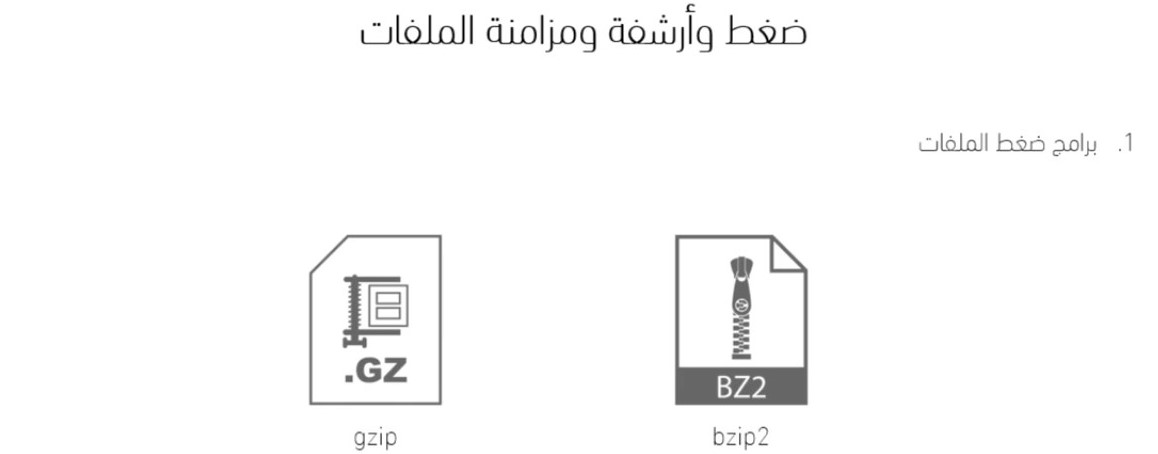
sudo apt update && sudo apt upgrade

يمكننا تثبيت الحزم المحملة على الجهاز بهذه الطريقة



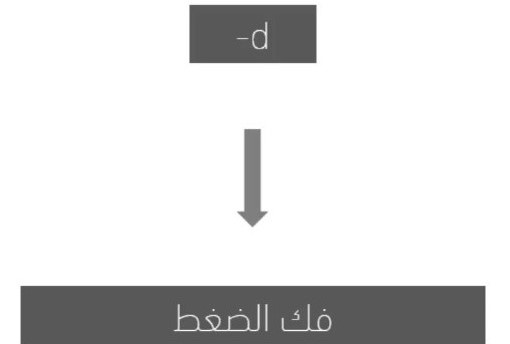
نكتب مكان : package\_file اسم الحزمة

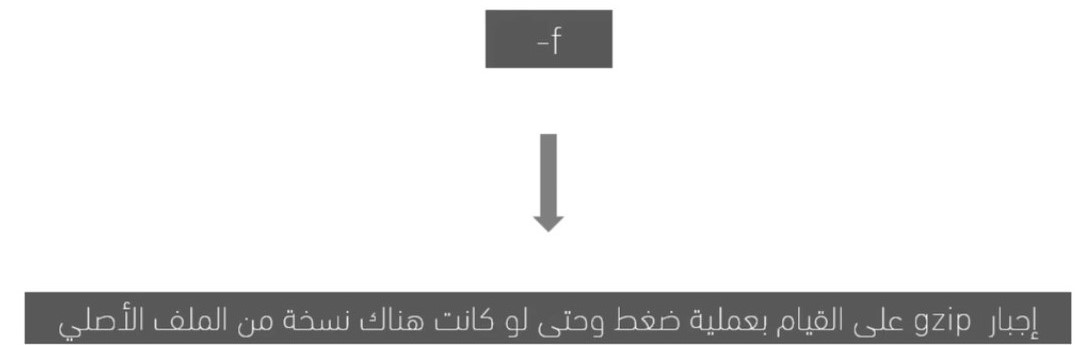
ضغط الملفات

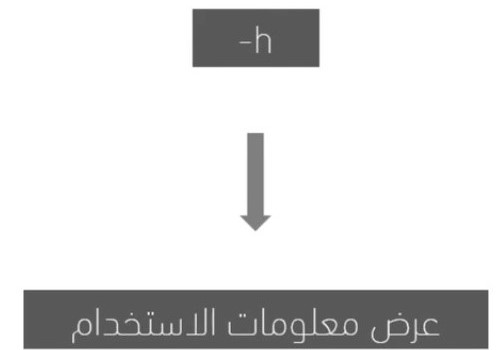




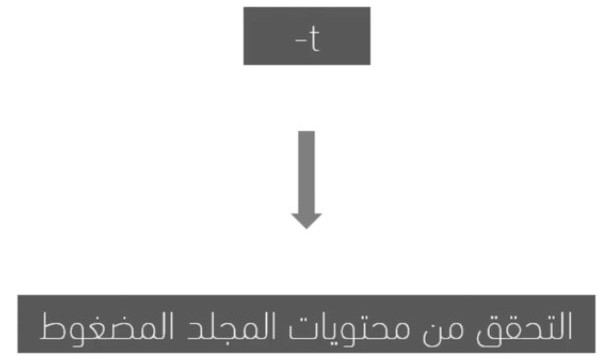
خيارات الأمر gzip

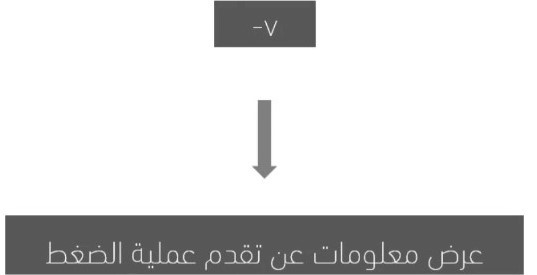


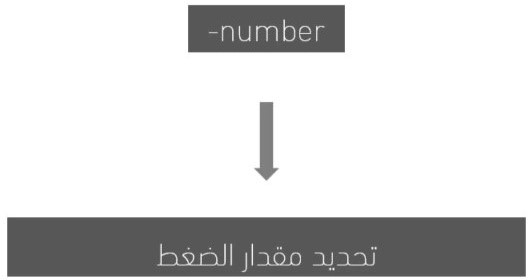




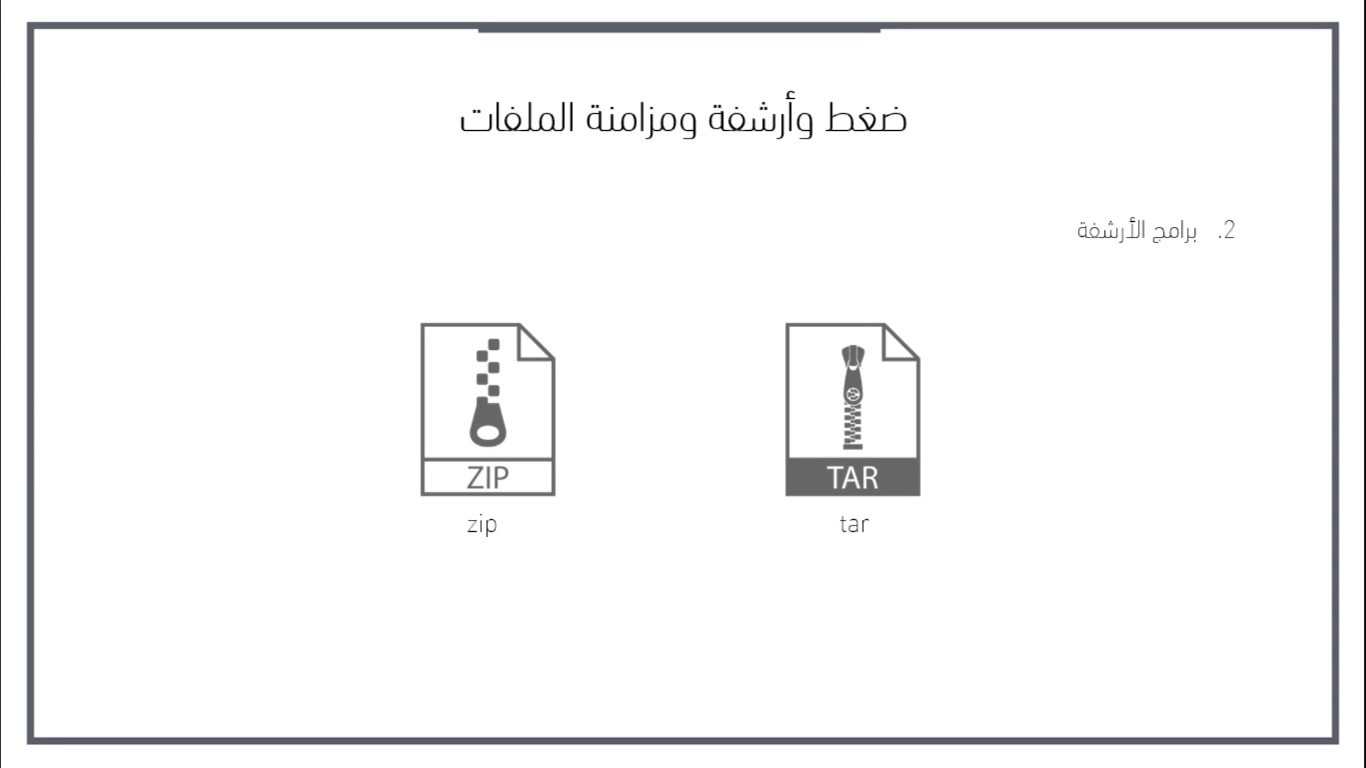








أرشفة الملفات







tar cf files.tar files ايش فائدة حرف ال f

قال ChatGPT:

الأمر:

tar cf files.tar files

يتكوّن من:

tar → برنامج ضغط/أرشفة الملفات.

c → اختصار لـ create، أي إنشاء أرشيف جديد.

f → اختصار لـ file، ويعني أن اسم ملف الأرشيف سيُحدَّد مباشرة بعده (في هذا المثال: files.tar).

🔹 فائدة حرف الـ f:

يخبر tar أن المخرجات (الأرشيف الناتج) يجب أن تُكتب في ملف (وليس إلى مخرج قياسي مثل الشاشة).

بدون -f، سيحاول tar إرسال الأرشيف إلى المخرجات القياسية (stdout) أو إلى جهاز شريط (tape device) في الأنظمة القديمة.

مثلاً:

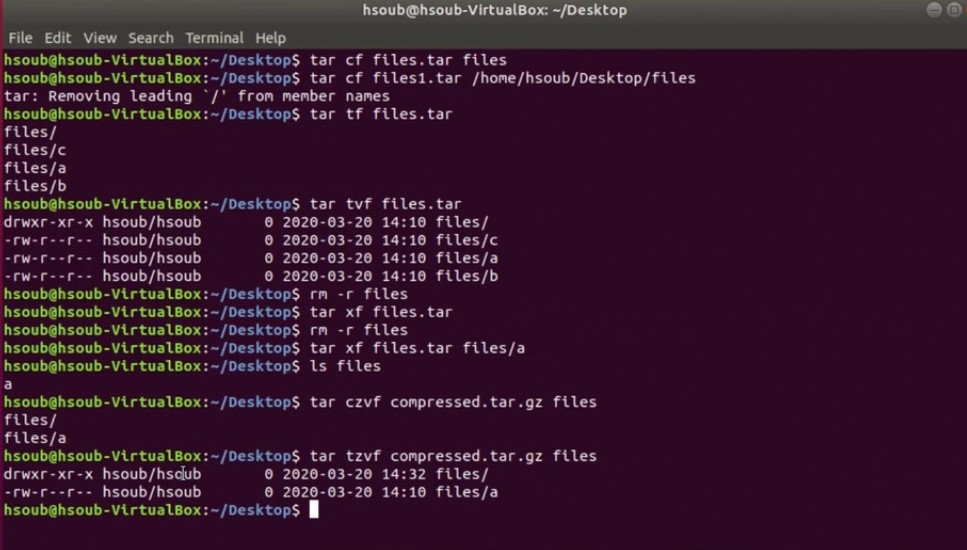
tar cf backup.tar myfolder هذا صحيح ✅

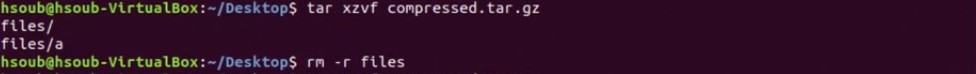
وهذا خطأ ❌ (لأن tar لا يعرف إلى أين يكتب الملف)

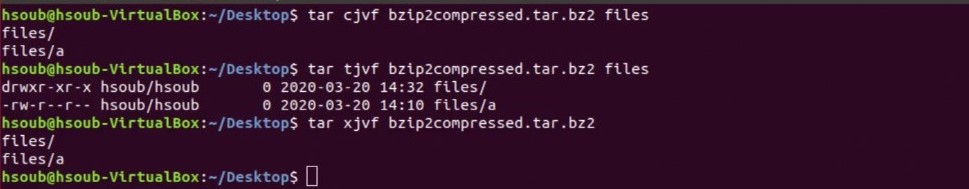
tar c backup.tar myfolder

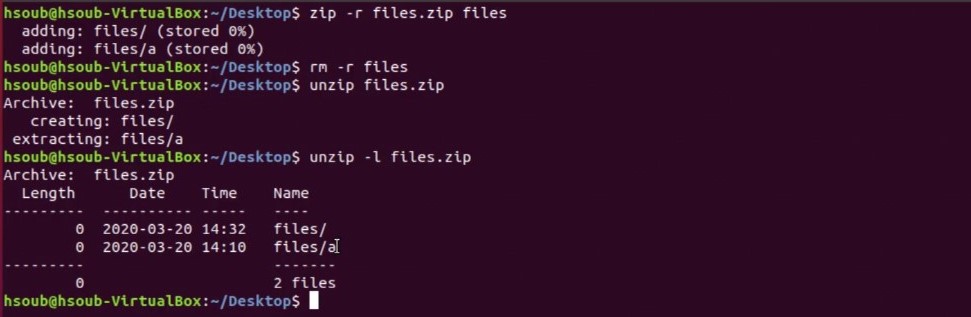
💡 باختصار:

حرف f يُستخدم لتحديد اسم ملف الأرشيف الناتج أو المدخل في أوامر tar.









ملاحظة :يفيد الأمر زيب نقل اللملفات بين أنظمة التشغيل المتختلفة لأنه متوافق معها

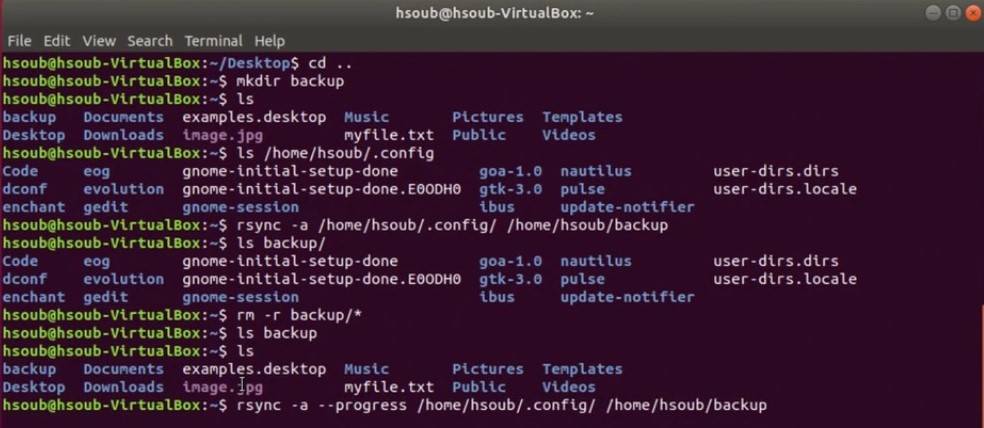
رابط فيه كل أوامر لينكس

[https://www.geeksforgeeks.org/linux-commands-cheat-sheet](https://www.geeksforgeeks.org/linux-commands-cheat-sheet/)

النسخ الاحتياطي

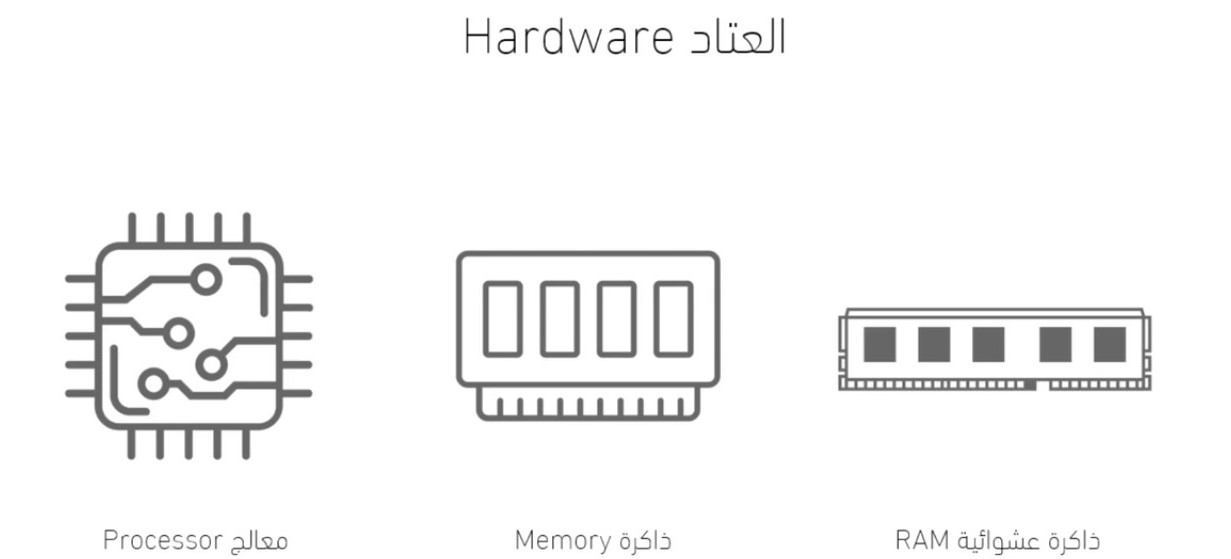


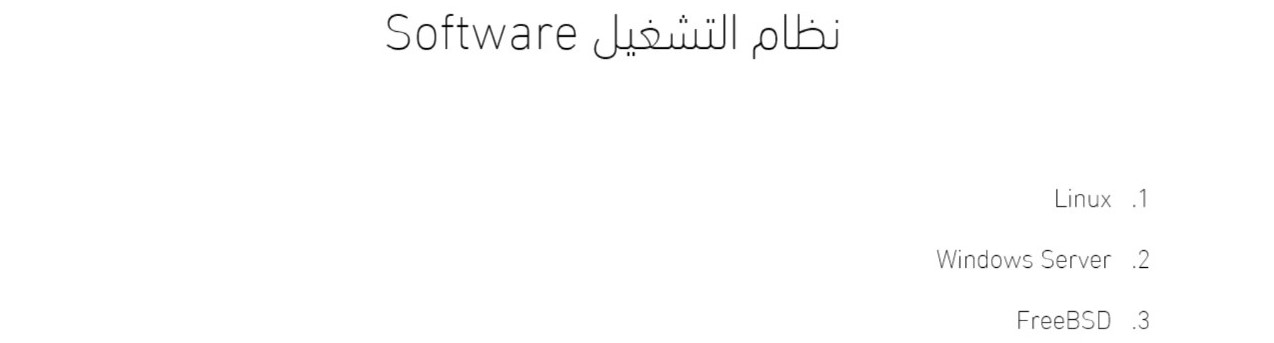




التعرف الخوادم

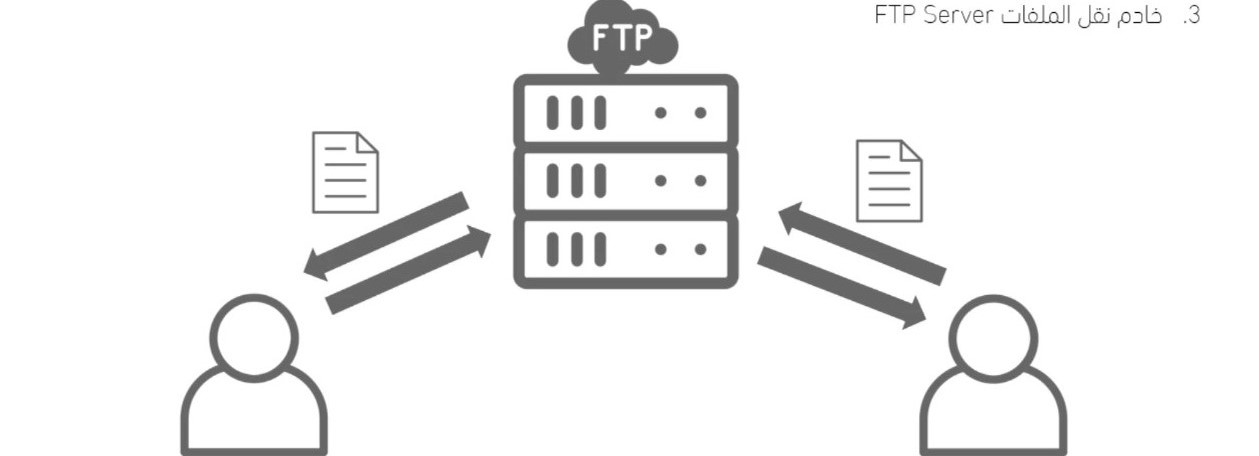


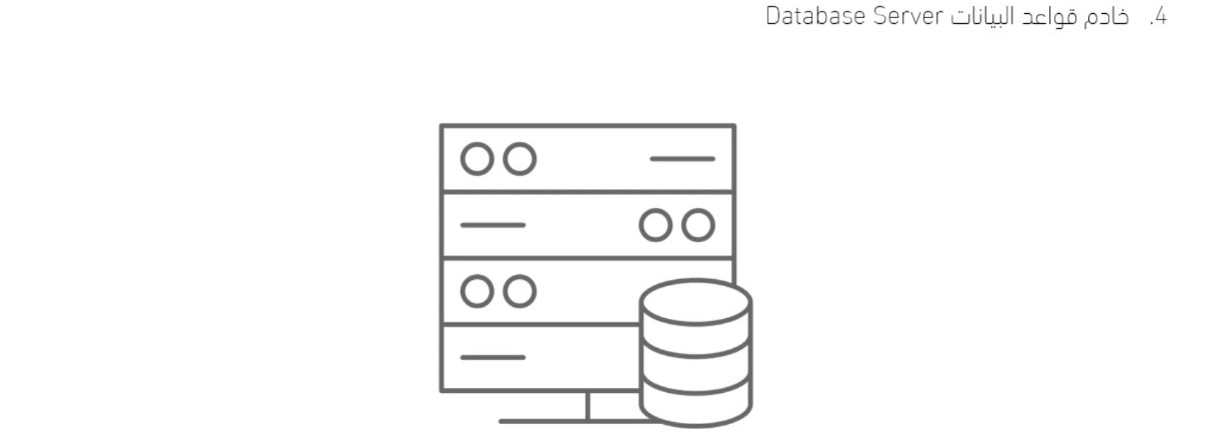












1️⃣ الفكرة الأساسية

NAS — Network Attached Storage

تخيل أن لديك خزانة ملفات مشتركة في مكتب، كل الموظفين يمكنهم الوصول إليها عبر الشبكة لوضع أو أخذ الملفات.

هي مساحة تخزين مركزية على الشبكة.

الأجهزة ترى الملفات كما لو كانت مجلدًا عاديًا مشتركًا.

مثال: تريد أن يشارك كل موظف المستندات، الصور أو الفيديوهات، بدون نسخ كل شيء على جهازه.

الأساسيات:

على مستوى الملفات، أي أنك تتعامل مع مجلدات وملفات.

بروتوكولات شائعة في لينكس: NFS (لأنظمة لينكس/يونكس)، SMB (للتوافق مع ويندوز).

المميز: سهل الاستخدام، مثل مشاركة مجلد عادي على جهازك.

SAN — Storage Area Network

تخيل أن لديك مجموعة أقراص صلبة كبيرة جدًا في مكان مخصص، ولكن بدل أن تصل إليها عبر المجلدات، فإن كل خادم يرى هذه الأقراص وكأنها أقراص داخلية في جهازه نفسه.

هي شبكة لتوصيل أقراص التخزين مباشرة بالخوادم.

الخوادم تستطيع إنشاء نظام ملفات عليها بنفسها، أي تتحكم فيها بالكامل.

مثال: قواعد بيانات كبيرة، سيرفرات افتراضية تحتاج سرعة عالية جدًا للوصول للتخزين.

الأساسيات:

على مستوى الكتل (Block Level) أي التعامل مع الأقراص كما لو كانت جزءًا من جهازك.

بروتوكولات شائعة: iSCSI (تخزين عبر الشبكة العادية)، Fibre Channel (شبكة سريعة جدًا).

المميز: أداء عالي جدًا، مناسب للأعمال الكبيرة والمعقدة.

2️⃣ الفرق البسيط بينهم

|  |  |  |
| --- | --- | --- |
| النقطة | NAS | SAN |
| مستوى التخزين | ملفات (File Level) | كتل/أقراص (Block Level) |
| طريقة الوصول | مجلدات مشتركة على الشبكة | خوادم ترى الأقراص كما لو كانت محلية |
| الأداء | متوسط | عالي جدًا |
| التعقيد | سهل | معقد |
| الاستخدام | مشاركة ملفات للمستخدمين | قواعد بيانات، سيرفرات، تطبيقات كبيرة |

تشبيه سريع لتسهيل الفهم

NAS = خزانة ملفات مشتركة في المكتب: تضع فيها كل الملفات ويمكن للجميع فتحها، إضافة أو قراءة المستندات.

SAN = مخزن أقراص كبير متصل بالخوادم: كل خادم يرى الأقراص وكأنها جزء من جهازه، يستخدمها لتطبيقاته مباشرة.

✅ الخلاصة

NAS مناسب للمشاركة السهلة للملفات.

SAN مناسب للأداء العالي والتطبيقات الكبيرة.

كلاهما في النهاية مساحات تخزين على الشبكة، الفرق فقط في طريقة الوصول والتحكم والأداء.

الاستضافة

**ما هي الاستضافة (Hosting)؟**

**الاستضافة هي ببساطة مكان على الإنترنت تخزن فيه موقعك أو تطبيقك أو ملفاتك حتى يمكن للناس الوصول إليها من أي مكان.**

**تخيّل الأمر مثل:**

**لديك متجر صغير، تحتاج محل لتعرض فيه بضاعتك.**

**على الإنترنت، الاستضافة هي المحل، واسم الموقع هو عنوان المحل الذي يستخدمه الزوار للوصول إليه.**

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**🔹 أنواع الاستضافة الأساسية:**

**الاستضافة المشتركة (Shared Hosting):**

**يشترك عدة مواقع في نفس الخادم.**

**رخيص، سهل الاستخدام، مناسب للمواقع الصغيرة.**

**الاستضافة الافتراضية الخاصة (VPS Hosting):**

**كل موقع لديه قسم مستقل على الخادم.**

**أداء أفضل وتحكم أكبر.**

**الخوادم المخصصة (Dedicated Server):**

**الموقع يستأجر خادم كامل لنفسه.**

**أداء ممتاز ومرونة كاملة، لكنه أغلى.**

**الاستضافة السحابية (Cloud Hosting):**

**الموقع يستخدم موارد من عدة خوادم متصلة بالسحابة.**

**مرونة عالية، سرعة عالية، تتحمل ضغط الزيارات الكبيرة بسهولة.**

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**🔹 الفكرة الرئيسية**

**الاستضافة = مكان على الإنترنت لموقعك أو بياناتك، بحيث يمكن للناس الوصول إليه في أي وقت.**

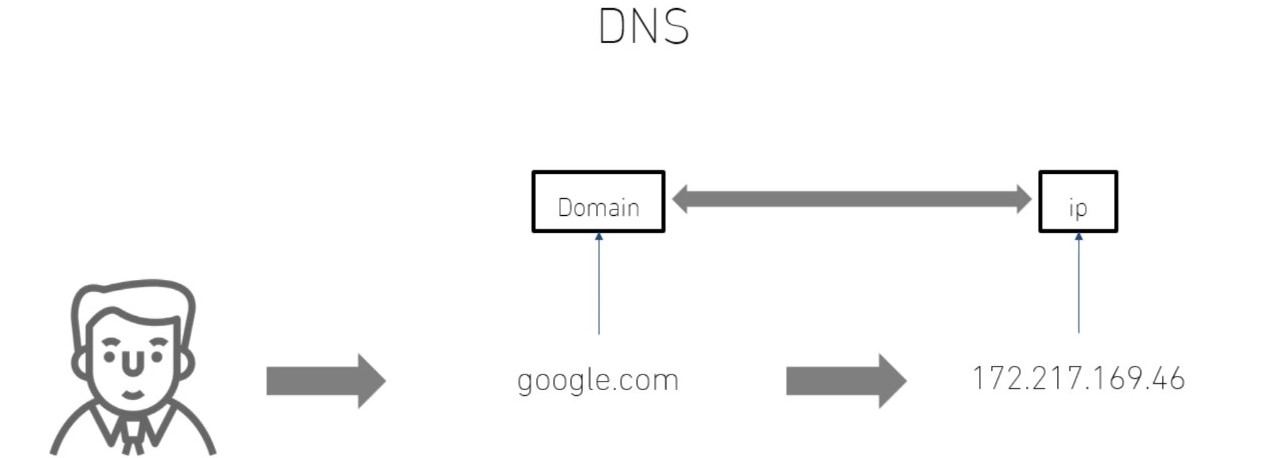
**إظهار معرف ip الخاص بموقع معين:**

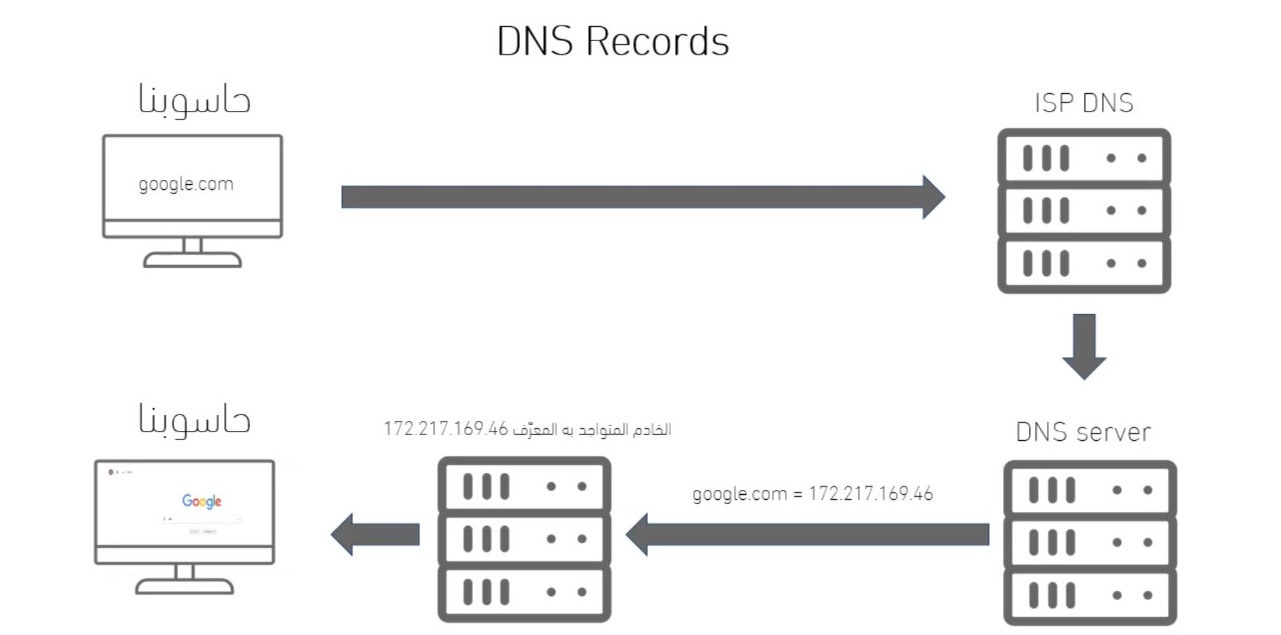
ping <website>

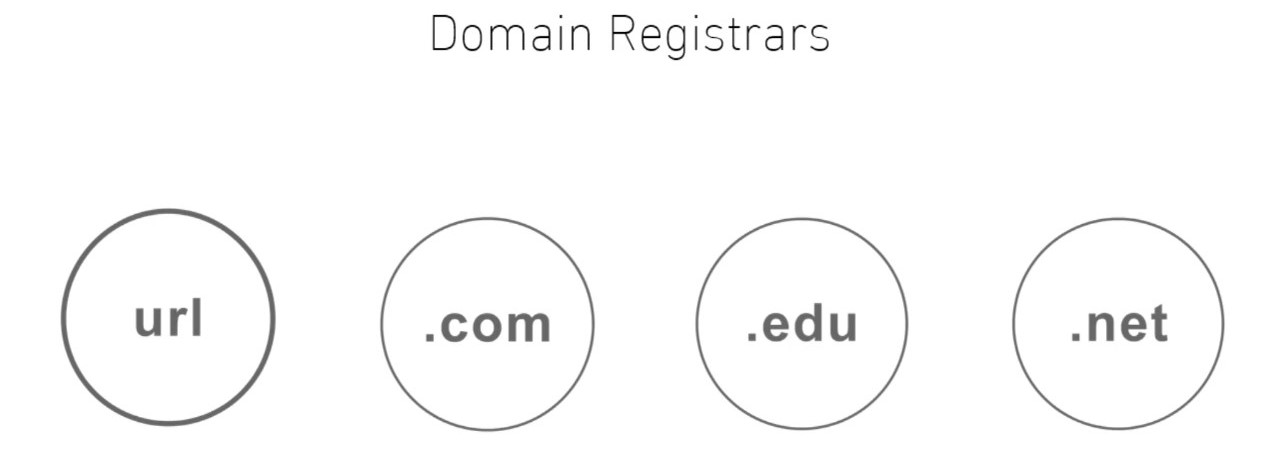
DNS











الدومين (Domain)

هو عنوان موقعك على الإنترنت.

مثال: google.com

يشبه عنوان بيتك: أي شخص يريد زيارة موقعك يحتاج يعرف هذا العنوان.

2️⃣ الاستضافة (Hosting)

هي المكان الذي يخزن فيه ملفات موقعك (HTML، صور، فيديو، قواعد بيانات…).

بدون استضافة، الدومين ما له فائدة لأنه لا يوجد موقع يظهر عند زيارته.

3️⃣ DNS (نظام أسماء النطاقات)

هو الدليل الذي يحوّل اسم الدومين إلى رقم IP الذي يعرفه الكمبيوتر.

بدون DNS، المتصفح لا يعرف أين يذهب عندما تكتب اسم الموقع.

4️⃣ سجلات الـ DNS (DNS Records)

هي تعليمات محددة في DNS تخبر الإنترنت أين يذهب.

أهمها:

A Record: يربط الدومين بالـ IP الخاص بالسيرفر.

CNAME: يربط دومين بدومين آخر.

MX Record: يحدد سيرفرات البريد.

TXT Record: لتأكيد الملكية وحماية البريد.

5️⃣ نطاق المستوى الأعلى (TLD مثل .com)

هو الجزء الأخير من الدومين، مثل .com, .net, .org.

يستخدم لتصنيف نوع الدومين: تجاري، شبكة، منظمة، حكومي، إلخ.

|  |  |  |
| --- | --- | --- |
| نوع النطاق الأعلى (TLD) | الامثلة | الوصف |
| gTLD (Generic TLD) | .com, .net, .org, .info | نطاقات عامة يمكن لأي شخص أو جهة تسجيلها. |
| sTLD (Sponsored TLD) | .edu, .gov, .mil, .museum | نطاقات مخصصة لجهات معينة أو فئات محددة، عادةً بحاجة لموافقة. |
| ccTLD (Country Code TLD) | .eg (مصر), .sa (السعودية), .us (الولايات المتحدة) | نطاقات رمزية لكل دولة، غالباً مرتبطة بالمواقع المحلية أو الخدمات الإقليمية. |
| new gTLDs | .tech, .store, .app, .xyz | نطاقات جديدة تم إطلاقها لتوسيع الخيارات beyond التقليدية (.com/.net). |
| Infrastructure TLD | .arpa | يستخدم لأغراض البنية التحتية للإنترنت (مثل التحويلات العكسية للعناوين). |

6️⃣ العلاقة بينه

الدومين = العنوان

الاستضافة = المكان الذي يخزن الموقع

DNS = الدليل الذي يربط الدومين بالاستضافة

سجلات DNS = التعليمات الدقيقة لكيفية الربط (IP، البريد، غيره)

TLD (.com) = تصنيف الدومين ضمن نطاق عالمي

💡 بشكل مبسط:

الدومين يشبه اسم بيتك، الاستضافة هي البيت نفسه، DNS هو دليل العناوين، وسجلات الـ DNS تحدد بالضبط أي غرفة أو باب يجب أن يفتح للوصول للموقع، وTLD مثل قسم الحي الذي ينتمي له بيتك.

إنشاء خوادم ويب على حاسوبنا

<https://academy.hsoub.com/courses/computer-science/operating-systems/servers/04-%D8%A5%D9%86%D8%B4%D8%A7%D8%A1-%D8%AE%D8%A7%D8%AF%D9%85-%D8%A3%D9%88%D8%A8%D9%88%D9%86%D8%AA%D9%88-%D9%88%D9%87%D9%85%D9%8A-%D8%B9%D9%84%D9%89-%D8%AD%D8%A7%D8%B3%D9%88%D8%A8%D9%86%D8%A7-r2497/>

الاتصال بالخادم عن طريق بروتوكول SSH





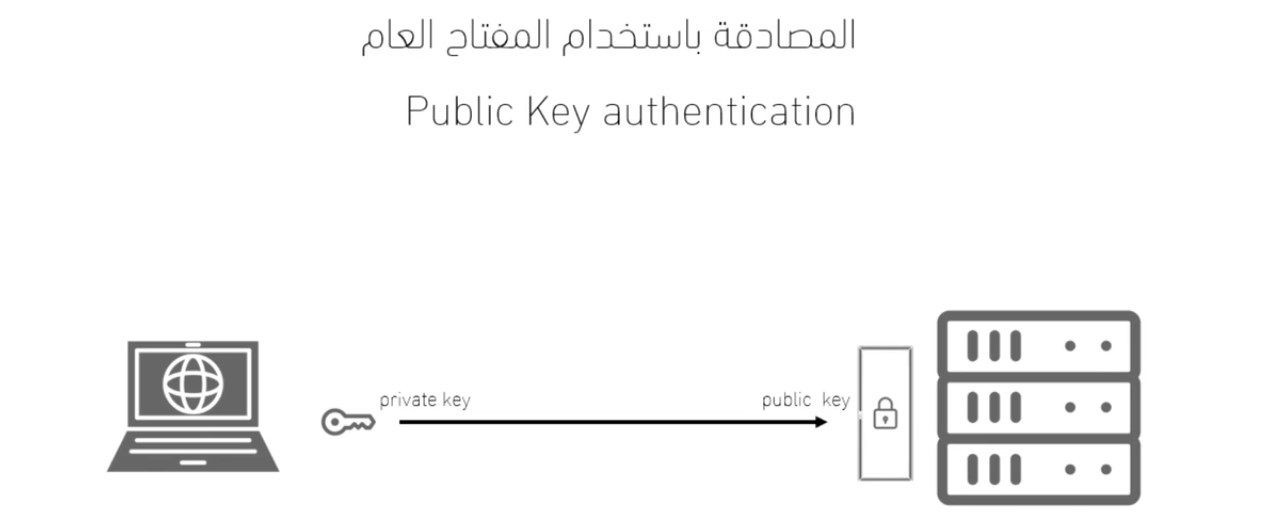
استخدام سطر الأوامر من نظام التشغيل ويندوز والاتصال بالخادم

عن طريق برنامج

PuTTY

طرق الاتصال بالخادم





تثبيت خادم vsftpd وضبطه على خادمنا الوهمي

اسم vsftpd هو اختصار لعبارة:

Very Secure File Transfer Protocol Daemon

🔹 شرح المعنى:

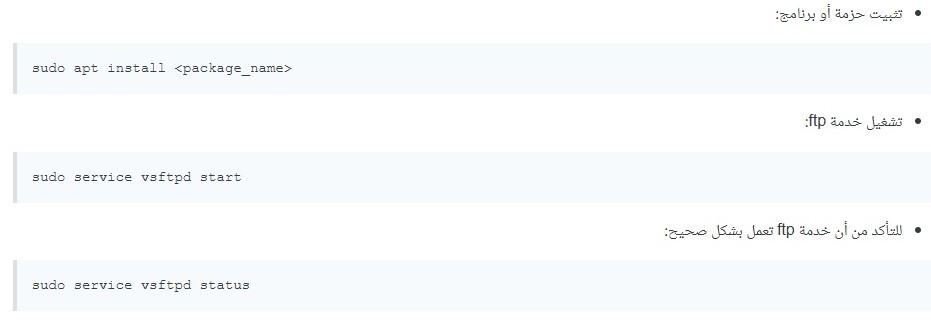
Very Secure = “آمن جدًا” — يشير إلى أن هذا الخادم صُمم مع تركيز كبير على الأمان.

FTP (File Transfer Protocol) = بروتوكول نقل الملفات — يُستخدم لنقل الملفات بين الأجهزة عبر الشبكة.

Daemon = “خدمة تعمل في الخلفية” على أنظمة لينكس/يونكس.

🔸 بمعنى آخر:

vsftpd هو برنامج خادم FTP يتميّز بالأمان والسرعة، ويُستخدم على نطاق واسع في أنظمة Linux لتقديم خدمة نقل الملفات عبر الشبكة.



إضافة مستخدمين ورفع الملفات على الخادم عن طريق FTP



