جامعة حمص
الكلية التطبيقية
قسم تقنيات الحاسوب نظم التشغيل الشبكية -1-
المحاضرة السابعة

تبادل البريد الإلكتروني

سنقوم بالتعرف على عدد من الطرق التي يمكن استخدامها لإرسال رسائل البريد الإلكتروني من خلال الطرفية. وتكمن أهمية هذه الأوامر لأنه من الممكن أن نحتاج إلى إرسال البريد الإلكتروني من خلال برامج السكربت التي تعرفنا إليها سابقاً أو برامج أخرى.

1) التعليمة sendmail :

ويتم ذلك عبر خطوتين. الخطوة الأولى هي إنشاء ملف نصي يحتوي على محتوى الإيميل الذي نريد إرساله. نقوم بتسمية هذا الملف مثلاً email.txt ويمكن لنا أن نستعرض محتوى هذا الملف باستخدام تعليمة cat كما يلي:

```
$ cat /tmp/email.txt

Subject: Terminal Email Send

Email Content line 1
Email Content line 2
```

الخطوة الثانية هي إرسال الإيميل وذلك كما يلي:

```
$ sendmail user@example.com  < /tmp/email.txt
```

2) التعليمة mail :

وهي التعليمة الأكثر شيوعاً لإرسال البريد الإلكتروني بحيث نقوم بتحديد العنوان الإلكتروني للمرسل إليه بعد التعليمة وبعده محتوى الإيميل.

لتحديد عنوان الإيميل نستخدم الرمز s– ثم نكتب عنوان الإيميل "Subject" الذي نريده

```
$ mail -s "Hello World" someone@example.com
```

ويمكن لنا إرسال محتوى الإيميل بشكل مباشر (دون قراءته من ملف)

```
$ mail -s "This is the subject" somebody@example.com <<< 'This is the message'
```

أو نستخدم تعليمة echo لتحديد محتوى الإيميل كما يلي:

```
$ echo "This is the body" | mail -s "Subject" -aFrom:Harry\<harry@gmail.com\> someone@example.com
```

أما إذا أردنا أن يتم تحديد محتوى الإيميل من خلال ملف نصي موجود لدينا سابقاً فنكتب:

```
$ mail -s "Hello World" user@yourmaildomain.com < /home/user/mailcontent.txt
```

مثال: سنقوم بكتابة سكربت لإرسال البريد الإلكتروني كما يلي:

```
#! /bin/bash
recipient="someone@example.com"
subject="Greetings"
message="Welcome to the Applied College"
' mail –s $subject $recipient <<< $message '
```

لاستقبال رسائل البريد الإلكتروني الواردة إلينا نستخدم تعليمة mail فقط كما يلي:

```
$ mail
Heirloom mailx version 12.5 6/20/10.  Type ? for help.
"/var/mail/enlightened": 7 messages 3 unread
 O  1 Enlightened        Sat Dec  6 11:33   21/658   This is the subject
 O  2 Enlightened        Sat Dec  6 11:34  773/25549 This is the subject
 O  3 Enlightened        Sat Dec  6 16:43   20/633   This is the subject
 O  4 Enlightened        Sat Dec  6 16:44   20/633   This is the subject
 U  5 Mail Delivery Syst Sat Dec  6 16:50   74/2425  Undelivered Mail Returned to Sender
 U  6 Enlightened        Sat Dec  6 16:51   19/632   This is mutts subject
 U  7 Enlightened        Sat Dec  6 16:52   19/647   This is mutts subject
?
```

عمليات الطباعة في لينوكس

الطريقة الأسهل للطباعة في لينوكس تتم باستخدام تعليمة cat نفسها بحيث نقوم بتوجيه الخرج إلى الطابعة التي نريد استخدامها. علماً بأن الطابعات في نظام لينوكس كغيرها من الأجهزة موجودة في المسار /dev مثلاً لو أردنا الطباعة إلى طابعة موجودة لدينا واسمها lp0 سنقوم بكتابة الأمر على الشكل التالي:

```
sudo cat myfile.txt > /dev/lp0
```

في هذه الحالة فإنه سيتوجب علينا تحديد اسم الطابعة في كل مرة (حتى لو كانت هي الطابعة الافتراضية الموجودة لدينا) وذلك لأن تعليمة cat تقوم بتوجيه الخرج افتراضياً إلى الشاشة وليس إلى الطابعة الافتراضية.

يمكن لنا الاستعاضة عن ذلك باستخدام الأمر lp وذلك كما يلي:

```
lp myfile.txt
```

كما يمكن لنا استخدام هذه التعليمة وتحديد اسم الطابعة التي نريد استخدامها باستخدام الوسيط d– وبعده اسم الطابعة مباشرة. مثلاً لو أردنا الطباعة إلى طابعة اسمها Sales نكتب:

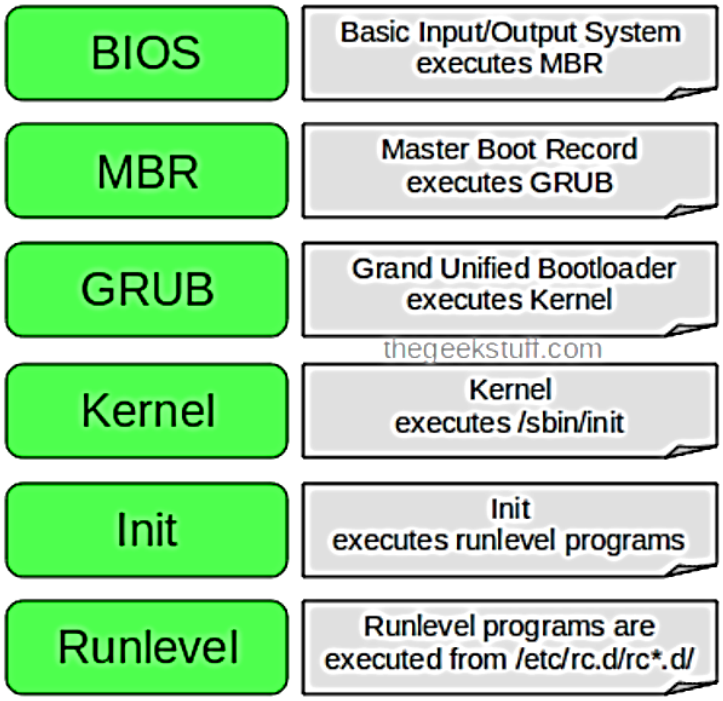
```
lp –dSales myfile.txt
```

لو أردنا أن نقوم بتحديد عدة ملفات لطباعتها كما يلي:

```
lp –dSales file1 file2 file3
```

إقلاع نظام التشغيل لينوكس

ويتم ذلك وفق ستة مراحل أساسية كما يلي:



أداة تحميل الإقلاع

نعتبر أداة تحميل الإقلاع Boot Loader هو المسؤول عن إقلاع نظام لينوكس وبدء تشغيله وفي البداية كان النظام المستخدم للإقلاع يسمى Linux Loader (LILO)

ثم تم تطوير عدد من أنظمة تحميل الإقلاع وأشهرها Grand Unified Bootloader (GRUB) و الإصدار الثاني منه حالياً GRUB2 يعتبر الآن أداة تحميل الإقلاع الأساسية المعتمدة لمعظم توزيعات نظام لينوكس الموجودة حالياً.

يقوم برنامج Bootloader بالعثور على نواة نظام التشغيل وتحميلها في الذاكرة. كما يمكن له أن يسمح للمستخدم بأن يتم الإقلاع باختيار النواة التي نريد استخدامها للإقلاع حيث يمكن لبعض توزيعات لينوكس أن تشتمل على أكثر من نواة. وتوفر لنا هذه الميزة إمكانية الإقلاع إلى إصدار نواة سابق ويمكن أن نلجأ إلى ذلك إذا فشل إصدار محدث لسبب ما أو إذا كان الإصدار الحديث غير متوافق مع برنامج ما فيمكن لنا في هذه الحالة أن نقوم بالإقلاع إلى إصدار نواة سابق لحل هذه المشكلة. يمكن لنا معرفة الإعدادات الخاصة بـ GRUB باستخدام الملف grub.conf والذي نجده في المسار التالي:

/boot/grub/grub.conf

أما النسخة الحديثة منه فنجد التكوين الخاص بها في الملف grub.cfg والذي نجده في المسار التالي:

/boot/grub2/grub.cfg

الوظيفة الأساسية لبرنامج Bootloader هي تحميل النواة Linux kernel في الذاكرة وتشغيلها. كلا الإصدارين من GRUB يعملان بشكل أساسي بنفس الطريقة ولهما نفس المراحل الثلاث ، وهذه المراحل هي:

1) المرحلة الأولى Stage 1

يقوم نظام الدخل والخرج الأساسي BIOS بالبحث في الأقراص الموجودة لدينا (القرص الصلب أو غيره) حيث يقوم بالبحث عن سجل الإقلاع boot record والموجود عادة في سجل التمهيد الرئيسي Master Boot Record (MBR) ثم يقوم بتحميل هذا السجل في الذاكرة ويبدأ بتنفيذه.

نظراً لأن سجل التمهيد يجب أن يكون صغيراً جداً (446 Bytes) فهو ليس ذكياً جداً ولا يفهم هياكل نظام الملفات.

والغرض الأساسي من هذه المرحلة هو تحديد موقع المرحلة التالية وتحميلها.

2) المرحلة الثانية Stage 1.5

القطاع الأول في القرص الصلب sector 0 هو الذي يحتوي سجل MBR بينما يبدأ أول قطاع منطقي في القرص الصلب يبدأ عند القطاع sector 63 فيبقى لدينا 62 قطاعاً فارغاً وهي التي يتم ملؤها في المرحلة الثانية حيث يتم تخزين ملف خاص بالإقلاع في هذه المنطقة الفارغة من القرص الصلب وهو الملف core.img والذي يحتوي على أنظمة تشغيل الملفات الشائعة مثل EXT وهو نظام الملفات القياسي في لينوكس وغيرها من أنظمة الملفات FAT,NTFS...

هنا يصبح نظام الملفات قادراً على تحديد موقع الملفات اللازمة للمرحلة التالية والتي سيجدها في المسار /boot

3) المرحلة الأخيرة Stage 2

يتم تحميل وحدات نواة نظام التشغيل والتي يجدها في المسار: /boot/grub2/i386-pc

وظيفة هذه المرحلة هي تحديد موقع نواة لينوكس وتحميلها إلى ذاكرة الوصول العشوائي RAM وتحويل التحكم بجهاز الكمبيوتر إلى النواة أي أن برنامج Bootloader سيقوم بإنهاء عمله وتسليم هذا العمل إلى النواة بعد أن تم تحميلها إلى ذاكرة الوصول العشوائي.

مستويات التشغيل Run Levels

بعد بدء نواة لينوكس سيقوم برنامج init بقراءة الملف inittab والموجود في المسار /etc/inittab وذلك لتحديد سلوك كل مستوى من مستويات التشغيل.

سيحاول النظام بدء مستوى التشغيل الافتراضي ما لم يحدد المستخدم غير ذلك.

يحتوي كل مستوى تشغيل على عدد معين من الخدمات التي تم إيقافها أو بدء تشغيلها، مما يمنح المستخدم التحكم في سلوك الجهاز.

توجد سبعة مستويات تشغيل، وهي مرقمة من صفر إلى ستة:

مستوى التشغيل Run Level	الوضع Mode	العمل Action
0	Halt	إيقاف النظام
1	Single-User Mode	لايقوم بتكوين واجهات الشبكة، أو بدء تشغيل البرامج، أو السماح بعمليات تسجيل الدخول لغير المستخدم root
2	Multi-User Mode	لايقوم بتكوين واجهات الشبكة
3	Multi-User Mode with Networking	يبدأ النظام بشكل طبيعي
4	Undefined	غير مستخدم / يقوم المستخدم بتعريفه
5	X11	مشابه للمستوى 3 مع تشغيل مدير النظام X
6	Reboot	إعادة تشغيل النظام

إيقاف التشغيل Shutting down

يقوم الأمر **shutdown** بإيقاف النظام بطريقة آمنة. عند القيام بإيقاف التشغيل سيتم إخطار جميع المستخدمين الذين تم تسجيل دخولهم بأن النظام في طريقه إلى التوقف. ولا يسمح بأي عمليات تسجيل دخول مرة أخرى. كما يمكن أن نقوم بهذه العملية بشكل فوري أو بتحديد وقت محدد نريد أن يتم إيقاف التشغيل فيه.

الصيغة العامة للأمر shutdown تكون على الشكل التالي:

shutdown [options] [time] [message]

- **options :** خيارات إيقاف التشغيل مثل halt أو power-off (الخيار الافتراضي عادة) أو إعادة التشغيل.
- **time :** وهو يعبر عن الوقت الذي يمكن أن نحدد من خلاله وقت تنفيذ عملية إيقاف التشغيل.
- **message :** وهو يعبر عن الرسالة التي سيتم بثها لجميع المستخدمين.

```
sudo shutdown
```

لو أردنا تحديد وقت محدد مثلاً عند الساعة 11:00 نكتب:

```
sudo shutdown 11:00
```

كما يمكن تحديد الوقت بصيغة أخرى، مثلاً لو أردنا أن تتم عملية إيقاف التشغيل بعد 10 دقائق من الآن نكتب:

```
sudo shutdown +10
```

يمكن أن نقوم بإرسال رسالة لجميع المستخدمين لنخبرهم بحصول إيقاف التشغيل وغالباً تحتوي هذه الرسالة على سبب حصول توقف النظام (مثل أن يكون السبب هو قيامنا بإجراء تحديث للنظام) كما يلي:

```
sudo shutdown +30 "Hardware upgrade"
```

أما لو أردنا أن نقوم بإعادة تشغيل النظام فإننا نستخدم الوسيط r– كما يلي:

```
sudo shutdown –r
```

وبنفس الأسلوب السابق يمكن أن نقوم بتحديد وقت أو رسالة للمستخدمين كما يلي:

```
sudo shutdown –r +5 "Updating Kernel"
```

في حال قمنا بجدولة إيقاف تشغيل مسبقاً لتتم في وقت محدد ثم أردنا إلغاء هذه العملية فنستخدم الوسيط c– كما يلي:

```
sudo shutdown -c "Canceling the reboot"
```

إدارة العمليات في لينوكس

في كل مرة نقوم فيها بتشغيل برنامج ما ضمن نظام لينوكس سيتم تحميل نسخة من هذا البرنامج في الذاكرة وعندها فإنه سيصبح مهمة قيد التشغيل. ونظراً لأن العمليات تستهلك موارد النظام مثل الذاكرة ووحدة المعالجة المركزية ومساحة القرص وغيرها لذلك فإن القدرة على إدارة العمليات تعد مهارة إدارية مهمة لمدير النظام. يتم تحديد حالة نظام لينوكس قيد التشغيل في لحظة معينة من خلال معرفة العمليات التي تعمل على النظام في تلك اللحظة.

عند بدء تشغيل النظام فإن أول برنامج سيبدأ تشغيله هو init ثم تبدأ بعد ذلك العمليات الأخرى تباعاً وتحصل كل عملية في نظام لينوكس على رقم تعريف خاص بها Process ID (PID) ونظراً لأن init هي دائماً أول عملية يتم تحميلها في الذاكرة فسيكون لها دائماً PID بقيمة 1.

أما Pid 0 فيشير عادة إلى العملية Swapper، وهي العملية التي تولد عملية init. يتم إنشاء العملية swapprt عند بدء تشغيل النظام فهي بالتالي أول عملية يتم إنشاؤها بواسطة النظام. يقوم swapper بإنشاء العملية init (PID=1) وهنا ينتهي عمل swapper وتصبح في حالة خمول ولذلك يطلق عليها اسم عملية الخمول idle process.

العمليات الخفية Daemon Process

العملية الخفية هي عملية يتم تشغيلها باستمرار على النظام أو تحميلها في الذاكرة فقط عندما يُطلب منا القيام بذلك. أثناء إقلاع لينوكس يتم تحميل العديد من العمليات الخفية في الذاكرة.

بعض عمليات البرامج الخفية الشائعة هي dhcpd (العملية الخفية الخاصة بتكوين بروتوكول DHCP) و lpd وهي اختصار لـ (Line Printer Daemon) وغيرها. عادة مايتم تحديد العملية الخفية بالحرف d في نهاية اسم العملية.

عمليات الأب والابن Parent and Child Processes

العملية الأب هي أي عملية تقوم ببدء أو إنتاج عملية أخرى. أما العملية التي تم بدؤها أو إنتاجها بواسطة العملية الأب فتسمى بالعملية الابن أو العملية الفرعية. تتحكم العملية الأصل في جميع عملياتها الفرعية.

يؤدي إيقاف العملية الأب إلى إيقاف العملية الفرعية تلقائياً.

بدء العمليات Starting a process

أسهل طريقة لبدء العملية هي كتابة اسمها في سطر الأوامر والضغط على Enter. وهنا من المهم لنا أن نعرف العمليات التي هي قيد التشغيل حالياً ولذلك نستخدم التعليمة ps والتي تقوم بعرض قائمة بسيطة عن أهم العمليات الجارية في الجهاز حالياً كما يلي:

```
alan@workstation:~$ ps
PID TTY          TIME CMD
23989 pts/0    00:00:00 bash
24148 pts/0    00:00:00 ps
```

أما لو أردنا معرفة كل العمليات التي هي قيد التشغيل حالياً نستخدم الوسيط e– ولعرض القائمة كاملة نستخدم الوسيط f– ويمكن أن ندمج هذين الوسيطين معاً فنكتب ef– كما يلي:

```
alan@workstation:~$ ps -ef
UID        PID  PPID  C STIME TTY          TIME CMD
root         1     0  0 Aug18 ?        00:00:10 /sbin/init splash
root         2     0  0 Aug18 ?        00:00:00 [kthreadd]
root         4     2  0 Aug18 ?        00:00:00 [kworker/0:0H]
root         6     2  0 Aug18 ?        00:00:00 [mm_percpu_wq]
root         7     2  0 Aug18 ?        00:00:00 [ksoftirqd/0]
root         8     2  0 Aug18 ?        00:00:20 [rcu_sched]
root         9     2  0 Aug18 ?        00:00:00 [rcu_bh]
root        10     2  0 Aug18 ?        00:00:00 [migration/0]
root        11     2  0 Aug18 ?        00:00:00 [watchdog/0]
root        12     2  0 Aug18 ?        00:00:00 [cpuhp/0]
root        13     2  0 Aug18 ?        00:00:00 [cpuhp/1]
root        14     2  0 Aug18 ?        00:00:00 [watchdog/1]
root        15     2  0 Aug18 ?        00:00:00 [migration/1]
root        16     2  0 Aug18 ?        00:00:00 [ksoftirqd/1]
alan     20506 20496  0 10:39 pts/0    00:00:00 bash
alan     20520  1454  0 10:39 ?        00:00:00 nginx: master process nginx
alan     20521 20520  0 10:39 ?        00:00:00 nginx: worker process
alan     20526 20506  0 10:39 pts/0    00:00:00 man ps
alan     20536 20526  0 10:39 pts/0    00:00:00 pager
alan     20564 20496  0 10:40 pts/1    00:00:00 bash
```

إرسال إشارات للعمليات Sending Signals

الإشارة هي ببساطة عبارة عن وسيلة لإخبار عملية بفعل شيء ما لا تفعله عادة. على سبيل المثال يمكن أن نقوم بقتل عملية ما حيث يمكن للمستخدم إرسال إشارة القتل إلى العملية Kill Signal. التعليمة المستخدمة لذلك هي تعليمة kill

```
kill –l
```

وكخرج لهذه التعليمة سيظهر لدينا قائمة بخيارات أمر القتل، وهي الإشارات التي يمكن استخدامها. بعض الخيارات الأكثر استخداماً هي:

SIGHUP	Hang Up Signal	تقوم بإعادة تحميل ملفات التكوين للعملية
SIGINT	Interrupt	المقاطعة
SIGQUIT	Quit	يقوم بمحاولة إنهاء العملية
SIGKILL	Kill	لإنهاء العملية بدون إمكانية حظر الإشارة أو تجاهلها
SIGTERM	Terminate	يتم إرسالها غالباً قبل إرسال إشارة القتل
SIGSTOP	Stop	لإيقاف العملية بدون إمكانية حظر الإشارة أو تجاهلها

نظام الملفات في لينوكس Linux File System

نظام الملفات في لينوكس هو بشكل عام طبقة مضمنة من نظام التشغيل لينوكس تستخدم للتعامل مع إدارة البيانات لتخزينها. يساعد ذلك في ترتيب الملف عند تخزينه على القرص الصلب. يقوم نظام الملفات بإدارة اسم الملف وحجم الملف وتاريخ إنشائه والمزيد من المعلومات حول الملف.

يشمل نظام الملفات في لينوكس ثلاثة أجزاء رئيسية:

- المجلد الجذر أو الدليل الجذر (/) The root directory
- تنسيق محدد لتخزين البيانات (أي نوع نظام الملفات مثل EXT3 و EXT4 و XFS وغيرها..)
- التقسيم partition أي وحدة التخزين المنطقية.

ويتميز نظام الملفات في لينوكس بالخصائص التالية:

1) **تحديد المسارات:** حيث يستخدم الشريط المائل للأمام (/) لتحديد المسار مثل /home/myfiles
2) **التقسيم والمجلدات ومحركات الأقراص:** لا يستخدم لينوكس الأحرف لتحديد أسماء محركات الأقراص وكل ما ينبغي علينا معرفته للوصول إلى الملف هو تحديد مساره بغض النظر عن التقسيمات.
3) **الحساسية لحالة الأحرف:** نظام الملفات في لينوكس حساس لحالة الأحرف فهو يميز بين الأسماء بالأحرف الصغيرة أو الكبيرة.
4) **امتدادات الملفات:** قد يكون للملف امتداد (مثل txt.) لكن ليس من الضروري أن يكون للملف امتداد أو لاحقة. وإذا كنا نستخدم مدير الملفات الرسومي فإنه يميز لنا رمز المجلدات عن الملفات حتى لو كانت بدون لاحقة.
5) **الملفات المخفية:** يقوم لينوكس بالتمييز بين الملفات العادية والملفات المخفية بحيث يقوم بتمثيل هذه الملفات بنقطة (.) قبل اسم الملف.

نظام النافذة الرسومية إكس X-Window System

غالباً ما يشار إلى النظام إكس على أنه بيئة العمل الرسومية الموزعة. وهو عبارة عن برنامج مفتوح المصدر، متعدد المنصات، ومبني وفق علاقة خادم لزبون client/server يوفر واجهة مستخدم رسومية Graphical User Interface (GUI) ضمن بيئة موزعة. ويتم هذا التوزيع باعتبار أنه يمكنك تشغيل النظام إكس لديك ليتم العرض على الشاشة في مكان ما، بينما التطبيق يعمل على جهاز كمبيوتر في مكان آخر.

النظام إكس شائع جداً بين مستخدمي لينوكس إلا أنه ليس منتجاً من قبل UNIX بل تم تطويره عام 1994 من قبل معهد ماساشوستس للتكنولوجيا وجامعة ستانفورد، وهو يعمل على مجموعة واسعة من الأجهزة باختلاف أنظمة التشغيل فيها.

مؤسسة X.org هي مجموعة مفتوحة تدير تطوير النظام إكس وتوحيد معاييره.

يطلق على هذا النظام اسم X Window System أو X11 أو فقط X ويوفر هذا النظام إطاراً أساسياً لبيئة واجهة المستخدم الرسومية GUI الخاصة به فهو يستخدم عناصر واجهة المستخدم الرسومية عند المستخدم لبناء طرق خاصة بالنظام. حيث تقوم هذه الطرق بإرسال تفاعلات المستخدم.

باستخدام مجموعات أدوات تطوير الواجهة الرسومية يمكن لمطور التطبيق إنشاء واجهة تطبيق جديد بشكل مخصص. ويمكن أن تكون هذه التطبيقات من نمط خادم وزبون server/client وذلك لأن X مبني على هذا النمط أساساً، وكذلك فهو مناسب لبناء تطبيقات تعمل عن بعد. يتيح لك هذا الأمر إمكانية العمل من جهاز الكمبيوتر الخاص بك مباشرة باستخدام تطبيق يعمل على كمبيوتر آخر. نستنتج أن نظام X مستقل عن الأجهزة التي يعمل بها ويمكن الاستفادة من ذلك لتشغيل تطبيقات ويندوز ضمن لينوكس. ومن ميزات هذا النظام من ناحية تعامله مع الشبكة:

- شفافية الشبكة Network Transparency
- القدرة الرسومية القابلة للتخصيص Customizable Graphical Capability
- القدرة على الارتباط بشبكات مختلفة The Ability to Link to Different Networks

بنية النظام X

1) نظام إكس يستخدم بنية الخادم والعميل client/server
2) الخادم الرئيسي لهذا النظام هو خادم X.org
3) الطرفية الأساسية عند المستخدم هي الطرفية X Terminal

تتواصل التطبيقات المختلفة مع الخادم وترسل الطلبات إليه وتتلقى المعلومات منه. كل ما تحتاج التطبيقات معرفته هو كيفية التحدث إلى الخادم مهما كان جهاز المستخدم الذي تعمل عليه.

يحتوي نظام X أيضاً على مكتبة تسمى Xlib تتعامل مع جميع اتصالات client/server منخفضة المستوى.

علاقة الزبون والخادم The Client/Server Relationship

- **X server :** يشير هذا المصطلح إلى برنامج يتم تشغيله عند مضيف محلي local host متصل بشاشة المستخدم وأجهزة الإدخال.
- **X client :** يشير هذا المصطلح إلى برنامج يعمل عند مضيف بعيد ويقوم بمعالجة البيانات أثناء الاتصال بخادم X.
- **Application server :** يشير إلى برنامج يعمل أيضاً عند مضيف بعيد ويقوم بمعالجة البيانات ويتواصل مع عملاء التطبيق.
- **Application client :** يشير إلى برنامج يتم تشغيله عند مضيف محلي متصل بشاشة المستخدم وأجهزة الإدخال.