

بسم الله الرحمن الرحيم

مقدمة عن الصدفة SHELL

ما هي الصدفة (shell)؟

كثيرا من المستخدمين لنظام لينوكس (Linux system) لا يعرفوا مفهوم كلمة الصدفة (shell) بالرغم أنهم لا يستطيعون التعامل مع لينوكس بدون هذه الصدفة (shell). وحتى تعرف أكثر عن هذه الصدفة (shell) وما أهميتها ما عليك فقط إلا أن تستمر في القراءة. تعتبر الصدفة (shell) حلقة الوصل بين المستخدم وبين قلب نظام التشغيل لينوكس (Linux kernel). حيث أن أي أمر تكتبه في المحرر (prompt) تقوم الصدفة (shell) بأخذ الأمر (command) وتفسيره ثم يمرر النتيجة إلى قلب النظام التشغيل لينوكس ((Linux kernel. وكما قلت أن الصدفة (shell) عبارة عن مفسرة للأوامر والصدفة (shell) تحتوي في الأساس على أوامر خاص بها، ويمكنك للصدفة أن تستخدم جميع البرامج والأدوات المتوفرة على النظام.

ولكي يكون الكلام أكثر دقة ، فالصدفة تستطيع أن تعتبرها macro processor ، بحيث أنك عندما تدخل لها سطر أمر فإنها تقوم بمعالجة هذا السطر بحيث أنها تقوم بتقسيم هذا السطر إلى عدة أجزاء وكل جزء من الأجزاء يتم ترجمته إلى تعليمات يفهمها نظام التشغيل . وقبل أن ندخل في التفاصيل يجب أن أنوه إلى شيء مهم وهو أنه بمجرد أن تضغط زر التشغيل الخاص بالحاسب فإن هذه العمليات تحدث بالترتيب لكي تتمكن من التعامل مع النظام أو (النواة) في حالتنا هذه، والعمليات بالترتيب هي كالآتي:

1- يقوم BIOS بتحميل أول مقطع (sector) في القرص الصلب ، حيث أنه يوجد برنامج تحميل النواة (LILO أو GRUB) ، يتم تحميله إلى الذاكرة الرئيسية RAM ومنه ثم يتم تنفيذ أي منهما ثم تظهر لك قائمة بها نظم التشغيل التي توجد بحاسبك ، طبعاً في حالتنا هذه سوف نختار نظام لينوكس (بغض النظر عن التوزيع المستخدمة).

2- يتم تحميل النواة من القرص الصلب إلى الذاكرة ، وهو عبارة عن ملف مضغوط يدعى image .

3- يتم فك ضغط هذا الملف ، ثم يتم تنفيذ الأوامر الخاصة ببدء العمليات المكلفة بإدارة الجهاز.

4- تقوم النواة بعمل فحص للأجهزة الموجودة وبدء تشغيلها .

5- تقوم النواة بتفعيل (mount) نظام الملفات الجذرية (root file system)، ونقل مثل /dev/hda .

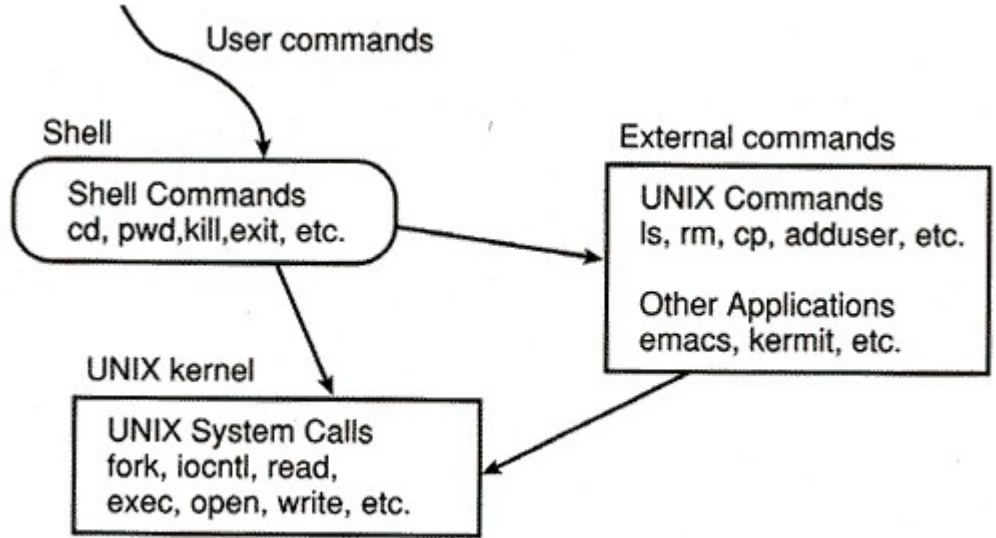
6- تقوم النواة بتنفيذ /sbin/init ، حيث أن هذا البرنامج أول برنامج تنفذه النواة ، لذلك فإن PID الخاص به يساوي 1 .

- 7- تقوم init بتنفيذ كل الملفات النصية التنفيذية (script) الخاصة بمستوى التشغيل (run level) ، سوف نطرق لمفهوم (run level) فيما بعد إن شاء الله.
- 8- تقوم init بإنتاج برنامج getty لكل طرفية (terminal).
- 9- getty هو المصنف الذي يظهر لك، ويطلب منك أن تدخل اسم المستخدم وكلمة المرور ، ومن ثم ينفذ الأمر /bin/login للتأكد من صحة المدخلات.
- 10- وفي النهاية يقوم الأمر login ببدء عمل الصدفة shell .
- وبهذا فأنت الآن دخلت إلى النظام وأصبحت مهيأ للعمل واستقبال الأوامر وتشغيل البرامج.
- نفهم من ذلك أن الصدفة تبدأ في العمل بمجرد دخولك للنظام وتظل كوسيط بينك وبين قلب النظام حتى تقوم بالخروج من النظام حيث ينتهي عمل الصدفة. وكل مستخدم في النظام له الصدفة الخاصة به ، والصدفة الخاصة بكل مستخدم تم تحديدها في ملف password الموجود في النظام (system password file) والموجود في المسار etc/passwd/ وهذا الملف يحتوي على عدة أشياء منها ID الخاص بكل مستخدم ونسخة مشفرة لكلمات المرور الخاصة بكل مستخدم واسم البرنامج الذي يعمل عليه المستخدم حالياً بعد دخوله إلى النظام.
- هذا الشكل يوضح لك موقع الصدفة shell في النظام.



كيف تعمل الصدفة؟

والشكل التالي يوضح كيف تقوم الصدفة (shell) بتفسير الأمر.



هذا الشكل يوضح الخطوات التي تتخذها الصدفة (shell) لتقرر ما الذي ستفعله مع الأوامر التي يكتبها المستخدم، وأول خطوة تقوم بها الصدفة (shell) أنه تحدد ما إذا كان الأمر الذي كتبه المستخدم من الأوامر الأساسية للصدفة أم لا، فإذا لم تكن من الأوامر الأساسية فإنها ترى ما إذا كانت من البرامج التطبيقية، والبرامج التطبيقية تنقسم إلى قسمين هما:

برامج خدمية خاصة بالنظام مثل ls, rm أو برامج تطبيقية للمستخدمين سواء كانت هذه البرامج تجارية أو عامة.

والصدفة (shell) تحاول في أن تجد هذه البرامج التطبيقية وذلك بالنظر في جميع الفهارس والتي توجد في مسار بحثك، وهذا المسار عبارة عن قائمة من الفهارس حيث توجد فيها البرامج القابلة للتنفيذ، وإذا لم يكن الأمر المدخل من أوامر الصدفة (shell) الداخلية أو ملفاً قابلاً للتنفيذ في المسار الذي توجد عليه فإنه سوف يظهر لك رسالة خطأ. أما إذا كان أمراً من أوامر الصدفة (shell) الأساسية أو ملفاً قابلاً للتنفيذ فإنه سوف يذهبها إلى system calls ثم يتم تمريرها إلى قلب النظام (system kernel).

وتتميز الصدفة بمفسر اللغة البرمجة (interpretive programming language) قوي، ولغة برمجة الصدفة

(shell programming language) تحتوي على معظم أدوات لغات البرمجة عالية المستوى مثل لاجمل التكرارية والدوال والمتغيرات والمصفوفات.

يمكنه أن يكتب أوامر وتستخدم معها الأدوات البرمجية الخاصة بالصدفة في ملف نصي عادي ، وهذه الأوامر تقوم مثلا بأخذ المعام الحيوية بالنسبة لك أو لبرامجك ، ثم تقوم بحفظ هذا الملف الذي كتبت فيه هذه الأوامر بامتداد sh ، وبالتالي يسمى هذا الملف بملف نصي تنفيذي (script). وفي كل مرة تنفذ هذا الملف يقوم بتنفيذ كل الأوامر التي كتبتها فيه بدون أن تكتبها أنت بنفسك مرة أخرى ، بمعنى أنه وفر عليك عناء كتابة عدة سطور قد تصل إلى الآلاف في كل مرة تريد تنفيذ فيها هذه المهمة .

قبل أن أكمل أحب التنويه على شيء مهم قد يكون غائبا أو غامضا للكثير منّا ، وهو عبارة عن سؤال ، ما هو الفرق بين سطر الأوامر الذي يكتبه أنا وبين الملف النصي التنفيذي؟ والإجابة هي أن سطر الأوامر لا تستطيع أن تقوم بالعمليات البرمجية ذات المستوى العالي ، مثل الجمل التكرارية ، لذلك فإننا نلجأ لحل مثالي وهو الملف النصي التنفيذي (script).

هل دراسة برمجة الصدفة (shell programming) شيء مهم؟

هذا السؤال من الأسئلة المهمة التي قد أو تواجه مستخدم أنظمة يونكس بشكل عام و لينوكس بشكل خاص. وإجابة هذا السؤال كالتالي:

بالنسبة لمدير النظام يجب أن يكون ملما ببرمجة الصدفة حتى يفهم تلك الملفات النصية التنفيذية ، حيث أن هذه الملفات مكتوبة بلغة الصدفة ، وبالتالي لا يستطيع مدير النظام أن يفهم هذه الملفات و لا كيف تتم سير العمليات في النظام إلا من خلال معرفته باللغة التي كتبت بها هذه الملفات. وأبسط مثال لذلك ، عندما يتم تحميل نظام لينوكس فإن النواة تقوم بتنفيذ الملفات النصية التنفيذية الموجودة في الفهرس /etc/rc.d ، حيث أنه من خلال هذه الملفات تستطيع أن تهئ النظام بالطريقة التي تناسبك ، وتستطيع أن تزود خدمات خاصة بك.

أما بالنسبة للمبرمجين وخصوصا محلي نظم التشغيل وخصوصا أنظمة يونكس وبالأخص جدا نظامي linux و GNU/hurd ، فبرمجة الصدفة بالنسبة لهم مهمة جدا لأنهم سيحتاجونها لعمل بعض العمليات التي قد تفيد برامجهم ، كأدوات مساعدة لتلك البرامج ، وأيضا لفهم البرامج الموجودة أصلا لكي يتسنى لهم تطوير هذه البرامج ، وأيضا لتحليل النظام نفسه ونتيج سير العمليات به لفهم طريقة عمله ولتطويره واستكشاف الأخطاء وإصلاحها ، وما إلى ذلك من الأمور البرمجية . ويجب التنبيه هنا ، بأنك تستطيع أن تغير محتويات هذه الملفات كما يحلو لك ، ولكنه احذر بأن تغير محتويات ملفات هامة قد تؤثر في تشغيل النظام نفسه.

متى لا نستطيع أن نستخدم الملفات النصية التنفيذية الخاصة بالصدفة؟

- في العمليات التي نريدها أن تنفذ بأكبر سرعة ممكنة والتي يعتبر الوقت فيها من العوامل الرئيسية المهمة، مثل عمليات الفرز والبحث و.....
 - في العمليات الرياضية المعقدة والتي تحتاج لدقة حسابية عالية .
 - في التطبيقات المعقدة ، والتي تطلب وجود لغات برمجية مثل لغة ++C / C ، حيث أنك تستطيع أن تقوم فيها بإنشاء دوال ذات صفات خاصة وتتيح لك أدوات برمجية أخرى ومهمة لا تجدها في الصدفة.
 - عندما نريد أن ننشئ تطبيقا يقوم بحماية النظام ضد عمليات الهجوم وما شابه ذلك .
 - عندما تحتاج أن ننشئ مصفوفة متعددة الأبعاد.
 - عندما تحتاج لبناء البيانات (data structure) .
 - ندما تحتاج إلى إنشاء واجهة رسومية أو حتى نريد أن نتعامل معها.
 - عندما نريد أن نتعامل مع الأجهزة الموجودة بالنظام بصورة مباشرة.
 - عندما تحتاج إلى port أو socket في عمليات الإدخال والإخراج.
 - عندما نريد أن نستخدم المكتبات.
 - عندما نريد أن ننشئ تطبيقا غير متاح المصدر، حيث أن الملفات التنفيذية النصية تكون دائما مفتوحة المصدر ولا يمكن خلقها.
- طبعاً ليست هذه كل الأسباب التي لا تجعلك أن تستخدم برمجة الصدفة ، ولكنها أشهر الأسباب ، بالإضافة ما تراه أنت أن لغة برمجة الصدفة له تستطيع أن تقدمه لك.

الصدفات الأكثر شيوعاً

- يوجد هناك عدة أنواع من الصدافات التي تعمل على نظام لينوكس و يونكس وكل من هذه الصدافات تمتاز بشيء من الأخرى وسوف نتعرض هنا لأنواع الصدافات المشهورة على مستوى النظام يونكس و لينوكس وسنقوم بالمقارنة بين هذه الصدافات ثم بعد ذلك نرشح لك الصدفة الأفضل عملاً واستخداماً... وفي النهاية فإن الأمر يعود لك في اختيار الصدفة الخاصة بك.
- وأكثر هذه الصدافات شيوعاً الصدفة Bourne shell واختصارها sh ، والصدفة C shell واختصارها csh والصدفة Korn shell واختصارها ksh . وكل من هذه الصدافات لها مميزات ولها عيوب.

الصدفة Bourne shell

قام بإنشائها Steven Bourne، وهي الصدفة الأصلية لنظام يونكس وتعتبر هذه الصدفة جيدة جدا في برمجة الصدفة تحت نظام يونكس ولكنها لديها قصور في التعامل مع المستخدمين مع الصدقات الأخرى.

الصدفة C shell

هذه الصدفة قام بإنشائها Bill Joy، وتتميز هذه الصدفة بسرعة استجابتها في التعامل مع المستخدمين وتفاعلاتهم، وتتميز أيضا بخاصية إتمام سطر الأوامر وهذا ليس موجودا في الصدفة Bourne shell. ولكه برمجة الصدفة تحت C shell ليست بمثل سهولة البرمجة تحت Bourne shell ولكنها تستخدم بكثرة مع طريق المبرمجين بلغة C لأن الصدفة C shell تتشابه مع لغة C مع حيث كتابتها لذلك فهي سميت بـ C shell.

الصدفة Korn shell

هذه الصدفة تم كتابتها مع طريق Dave Korn، حيث أنه أخذ مميزات الصدفتين Bourne shell و C shell وضمهم في صدفة وتعتبر هذه الصدفة ملائمة تماما مع الصدفة Bourne shell. والصدفة ksh صدفة فعالة وتمتلك واجهة تفاعلية جيدة وواجهة برمجية أيضا جيدة.

الصدفة BASH shell

كلمة bash اختصار لـ bourne again shell.

تتميز هذه الصدفة بأشياء بها كل مميزات الصدقات السابقة، بالإضافة بأن هناك مميزات أخرى تتميز بها. وهي الصدفة القياسية لأنظمة GNU كما أنها تعتبر أشهر وأهم صدفة في أنظمة يونكس بشكل عام.

وبالإضافة إلى هذه الصدقات يوجد هناك صدقات أخرى وتتميز بما تتميز بها الصدقات السابق ذكرها ومنه هذه الصدقات: tcsh وهي تعتبر امتداد للصدفة csh، والصدفة Public Domain Korn Shell واختصارها pdksh وهي امتداد للصدفة ksh.

يمكنك أن يحتوي نظامك على أكثر من صدفة، ولكي تعرف الصدفة التي تعمل أنت عليها إلا أن عليك أن تكون الأمر التالي:

```
echo $SHELL
```

وسوف تكون المخرجات إما sh أو csh أو ksh أو bash ، حيث أن sh تدل على الصدفة Bourne و csh تدل على صدفة C ، و ksh تدل على صدفة Korn و bash تدل على الصدفة bash .

لو كنت تعمل تحت صدفة معينة وأردت أن تنتقل إلى صدفة أخرى ما عليك إلا أن تكتب اسمها ، فلو أردت مثلا أن تنتقل إلى الصدفة bash اكتب الأمر bash ، ولو أردت الرجوع إلى الصدفة الأصلية ما عليك إلا أن تكتب الأمر exit . لو ظهر لك رسالة not found فهذا يدل على أن نظامك لا يحتوي على هذه الصدفة.

لو أردت أن تعرف أيه مكان الصدفة bash في النظام فكتب أحد الأوامر التالية: **whereis bash** أو **whence bash** أو **which bash** . أو اكتب الأمر التالي:

```
grep bash /etc/passwd | awk -F: '{print $7}' | sort -u
```

وسوف تكون النتيجة كالتالي:

```
/usr/local/bin/bash أو /bin/bash
```

كما علمت فإنه يمكنه أن يكون في نظامك أكثر من صدفة ، لنفرض أن الصدفة الافتراضية التي تعمل عندما تدخل إلى النظام هي C ، فهل يمكنك أن تجعل الصدفة bash هي الافتراضية بحيث أنك إذا دخلت إلى النظام تكون هي الصدفة الافتراضية بدلا من C ؟

نعم ، يمكنك ذلك باستخدام الأمر **chsh bash_name** ، حيث أن bash_name هي المكان أو المسار التي توجد فيه الصدفة والتي نتجت من تنفيذ الأمر الذي كتبناه منذ قليل :

```
grep bash /etc/passwd | awk -F: '{print $7}' | sort -u
```

وبالتالي سيكون الشكل النهائي للأمر chsh كالتالي:

```
% chsh /usr/local/bin/bash
```

أو

% `chsh /bin/bash`

إذا لم توجد صدفة `bash` فسوف تظهر لك رسالة خطأ.
