كتابة الشل السكربت Shell Scripts في الـ لينوكس

مقدمة:
يعتبر الـ shell أحد أهم مميزات النظام لينوكس ويعتبر قوة كبيرة بالنسبة له، وذلك لإمكانياته العالية جداً. حيث يمكنك من خلاله عمل الكثير من العمليات والتطبيقات على الملفات، والتحكم الكامل بالنظام من خلاله. يعتمد الـ shell على سطر الأوامر، أي إنه يستقبل الأوامر على شكل سطور تكتب عليه ويقوم هو بتفسيرها ومن ثم تنفيذها. في كتابي هذا سيكون الشرح مخصص للـ shell التي نوعها BASH. هذه الـ shell لها ميزات كثيرة وقوية جداً منها، إنك تستطيع أن تضع مجموعة سطور من الأوامر داخل ملف وتحفظه ومن ثم تقوم بتشغيل هذا الملف سيعمل وكأنه برنامج. هذا الملف هو ما يسمى بـ shell script.

هناك عدة دوافع لتعلم الـ shell scripting أي كتابة السكربتات:
1- تستطيع من خلاله أتمتت العديد من العمليات اليومية التي تقوم بها مثل أخذ نسخ إحتياطية للقاعدة بيانات. ولهذا ستسهل عليك عملك وتجعل عملك أكثر راحة بدل من القيام بهذه المهمات بشكل يدوي يومياً.
2- الكثير من العمليات التي يقوم بها النظام وعمليات الإقلاع Booting للنظام يتم التحكم بها من خلال Shell Scripts. فإن كنت تريد أن تعرف كيف تعمل أو تعدل عليها وعلى بعض من ما فيها، فعليك أن تفهم كيف تعمل هذه السكربتات وماذا يعني محتواها اولاً.
3- تعلم كتابة الـ Shell Scripts أسهل بكثير من تعلم أي لغة برمجة أخرى.
4- الـ Shell Scripts بإمكانك تطبيقها تقريباً على أي نظام nixWare.

النقاط السلبية الوحيدة في هذه اللغة هي:
1- الـ Shell Scripts نوعا ما بطيئة مقارنة مع لغات البرمجة الأخرى. يرجع هذا الى إنها تقوم لا تقوم بتحويل النصوص المدخلة في السكربت الى Binary Code جاهز للتنفيذ.
2- الـ Shell Scripts ممكن تستهلك من المعالج CPU الكثير من قدرته أثناء تنفيذها.

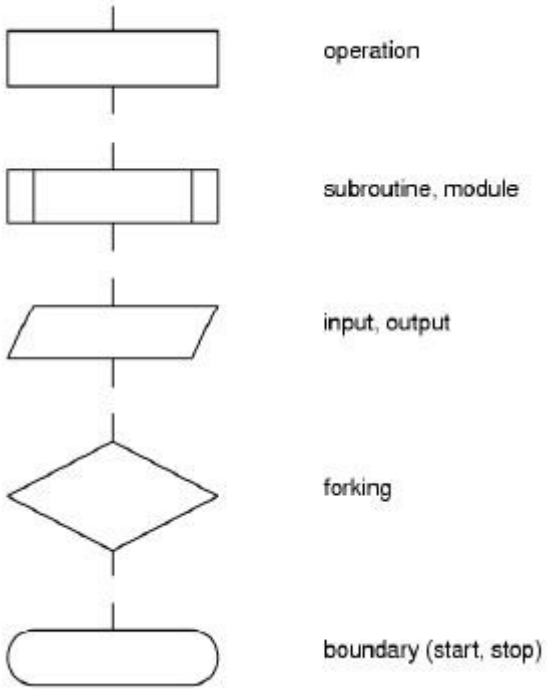
كما هو معروف لينوكس يقدم العديد من الـ Shell ولهذا السكربت الذي تعمله لـ shell معيين يمكن تشغيله على shell آخر ولكن ليس مظموناً دائماً. هذا الكتيب الصغير يهدف الى شرح كيفية تعلم كتابة السكربتات على الـ shell الذي نوعه BASH. سيتظمن الكتاب هذا أمثلة على كل خطوة تتعلمها في عالم كتابة السكربتات، ولهذا ما عليك سوى تطبيقها ومن ثم تحاول أن تقوم أنت بممراسة هذه المسألة وبأفكار جديدة.

طريقة التعلم لكتابة السكربتات:
سأقوم بالبدأ بمثال بسيط جداً يتظمن سطر واحد من الكود ومع إزدياد كمية المعلومات المأخوذة في كيفية كتابة السكربتات سنطور السكربت لتلبيت وفهم هذه المعلومات. سأقوم بأخذ مشروع بسيط طلب مننا لعمله من خلال سكربت. هذا المشروع هو نظام متابعة الطلبات الذي يمكن أن يستعمله الأدمن لمتابعة هذه الطلبات. حيث سيكون لدي في الأخير أداة أقوم بتشغيلها من سطر الأوامر، وتقوم بفتح الميل الخاص بالأدمن الذي إسمه "binary" وتقوم بعرض هذه الطلبات وأيضاً يمكن حذف ما تشاء منها.

طريقة تعلم كتابة السكربتات:
لتعلم كتابة السكربتات هناك عدة طرق وتختلف من كتاب وكاتب الى آخر، ولكن ما سأقوم بذكره هنا هي نفس الطريقة التي تعلمت فيها البرمجة في لغة البرمجة باسكال PASCAL في سنة أولى جامعة (العراق - بغداد). الطريقة التي أراها فعلاً مناسبة للتعلم هي من خلال معرفة المخطط الإنسيابي لتنفيذ السكربت وكيف سينفذه. هذه تتمثل بمعرفة الـ Flow Chart الخاص بالسكربت وأيضاً نتعلم كيف يمكن رسمه.

فوائد إستعمال Flow Chart في كتابة السكربتات:
1- تجبر الكاتب على وضع الخطوات التي يجب أن تقوم السكربت بإتباعها لكي تصل الى الهدف المطلوب. حيث ستسهل هذه الطريقة ال
2- توضح بشكل ممتاز الخوارزمية Algorithm المستعملة في سير عملية تنفيذ هذه السكربت.

أليك الرموز الأساسية في كتابة المخطط الإنسيابي Flow Charts



القواعد الأساسية في كتابة الـ Shell Scripts:

قبل أن ندخل في كتابة أول سكربت، هناك قواعد أساسية يجب أن نقوم بتذكرها دائماً. وهي:

1- الـ Shell Script هي عبارة عن ملف نصوص من نوع ASCII يحتوي على أوامر لتنفيذها بشكل متسلسل.

2- لتنفيذ هذه السكربت يجب أن يكون للمستخدم الذي سيقوم بتنفيذها له صلاحيات r قراءة و x تنفيذ عليها. يجب أن تقوم بإعطاء هذه الصلاحية للمستخدم لأنه النظام لا يسمح بتنفيذها عادة بشكل أساسي. لعمل ذلك قم بما يلي : chmod +x script.sh

3- بإمكانك تشغيل السكربت من shell آخر مثل: sh script.sh. هنا ليس شرط أن يكون السكربت له صلاحيات x تنفيذ عليه.

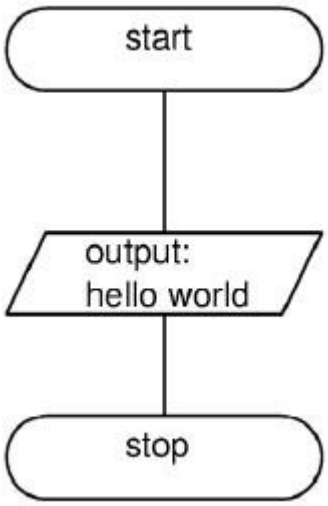
4- إذا كنت تريد تنفيذ هذه السكربت بغض النظر عن المجلد الذي أنت فيه حالياً current working directory، كل ما عليك فعله هو أن تضيف مسار المجلد الى المتغيير PATH الذي فيه المسار لجميع الملفات التنفيذية. في SuSE يقومون بعمل مجلد إسمه bin بداخل المجلد الرئيسي للمستخدم ويقومون بإضافة هذا المجلد الى المتغيير PATH كمايلي:

```
export PATH=$PATH:~/bin
```

إن لم تقم بعمل هذه الخطوة فإذن في كل مرة تريد تنفيذ هذه السكربت يجب عليك أن تنفذها من مسارها الأصلي.

5- عندما تقوم بتسمية السكربتات، ستكون فكرة جيدة لك أن تضع الـ sh. بعد الأسم لهذا الملف لكي تستطيع أن تمييزه على إنه شل سكربت. إن لم تقم بذلك مثلاً وكنت تريد أن تعمل سكربت إسمها less فهنا كيف ستعرف أي ملف تقوم بتنفيذه ؟ خاصة إن كنت قد أضفته الى مسار الملفات التنفيذية. ولهذا يفضل إنك تعمله less.sh لكي تميزه عن الأمر less الأصلي.

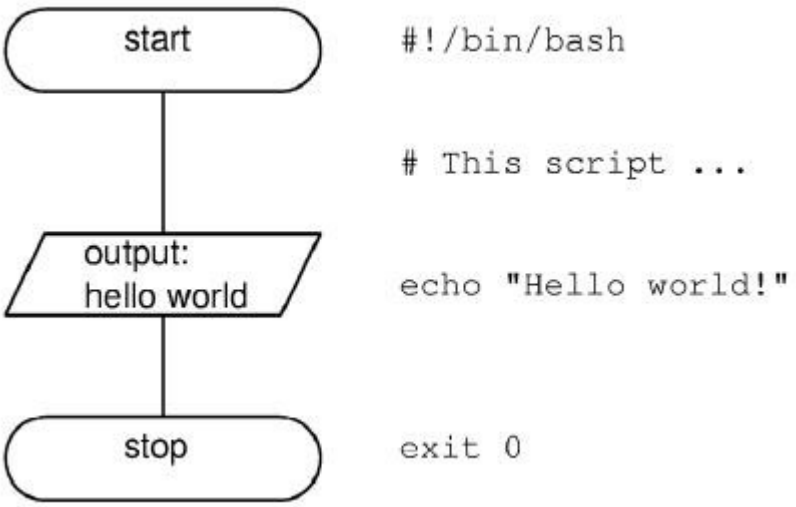
أول وأبسط سكربت:
أبسط سكربت هو الذي يقوم بطباعة Hello World. لم أرى كتاب لتعليم لغة البرمجة إلا وبدأ بهذا السكربت على إنه أول مراحل التعلم للبرمجة في تلك اللغة.



حيث تتظمن هذه السكرب على:
1- بداية السكربت.
2- الأمر الذي يقوم بطباعة جملة Hello World.
3- نهاية السكربت.

الكود:

```
#!/bin/bash
#B!n@ry 1st Script
echo "Hello World"
exit 0
```



مسألة مهمة هي إنه هذه الخطوات يمكن تطبيقها على أي سكربت بغض النظر عن حجمها وعملها. كلهم لهم بداية ولهم نهاية وبالوسط أو بينهما يوجد الأوامر المطلوب تنفيذها وهو الأمر المختلف من واحدة الى آخرى. الآن

لنحلل السكربت الأخير. السكربت يتظمن التالي:
- البداية: أول سطر يجب دائماً أن يكون ما يسمى بالـ shebang مثل:

```
#!/bin/bash
```

حيث يقوم هذا السطر بتحديد الشل الذي ستعمل عليه هذه السكربت. طبعاً كأي برنامج آخر سيقوم بتنفيذ هذه السكربت في Sub Shell. أيضاً يفضل أن تضع في بداية السكربت وصف لما تقوم به هذه السكربت. لكي تقوم بذلك فقط ضع قبل الوصف الإشارة #. أيضاً يعتبر من الأمور المهمة أو المرغوب بها بشدة في عالم البرمجة هي أن تذكر إسم كاتب السكربت، تاريخ كتابتها، ورقم النسخة. أيضاً إن كانت ستحتوي على متغييرات أو دوال Functions يفضل أن تقوم بكتابتهم في البداية.

- الأوامر: المثال في الأعلى كان يتظمن الأمر echo فقط والذي كان مسؤولاً عن طباعة الجملة Hello World. الأمر echo سترى إنه مستعمل بكثرة في عالم الشل سكربت وذلك لغرض عرض معلومات على الشاشة.

- النهاية: قبل أن تنتهي السكربت، يفضل أن تقوم بتحديد القيمة التي ستقوم بإرجاعها السكربت على إنها أنتهت. هذه القيمة تعرف الـ process الأب بأي حالة أنتهت السكربت. لمعرفة قيمة الحالة التي أنتهت عليها السكربت قم بعمل ما يلي:

```
echo $?
```

أي سكربت تقوم بكتابته يجب أن يستعمل هذه الهيكلية البسيطة.

الآن لنقم بإضافة بعض الإمكانيات التي تتيحها لنا الأمر echo:

```
#!/bin/bash
#B!n@ry 1st Script
echo -e "\aHello\nWorld"
exit 0
```

سيقوم هذا السكربت بتفعيل خاصية الـ Backslash بوساطة الخيار e- والـ a\ سيقوم بإصدار صوت Beep من جهازك أي alert والـ n\ سيقوم بالنزول الى سطر جديد ومن ثم إكمال عملية الكتابة. إذن الناتج النهائي سيكون كالتالي:

```
rul3z:~ # ./shell.sh
Hello
World
```

بالطبع مع إصدار صوت beep من جهازك.

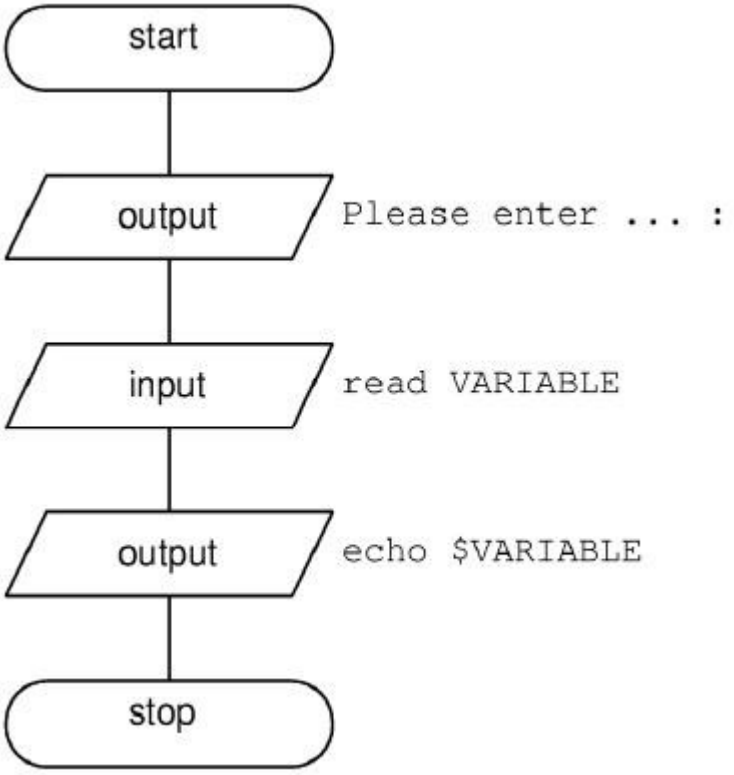
كيفية كتابة سكربت قابلة للقراءة من المستخدم:
لعمل ذلك نستعمل الأمر read داخل السكربت. هذا الأمر يتعامل مع متغيير Variable ويقوم بتخزين القيمة المدخلة في هذا المتغيير. بعد ذلك بإمكانك إستعمال هذا المتغيير للقيام بالعمليات المطلوبة على مدخلات المستخدم. مثال على ذلك:

```
read VARIABLE
```

هذا الأمر يقوم بقراءة المدخلات من المستخدم وتخزينها في متغيير إسمه VARIABLE. عندما يصل السكربت الى هذا السطر، يقف عن العمل بإنتظار المستخدم بالقيام بإدخال البيانات ومن ثم الضغط على Enter. طبعاً يفضل أن تطبع للمستخدم جملة معينة تعرفه ماذا يفعل، لأنه السكربت ستبقى هكذا متوقفة إن لم يقم المستخدم بإدخال بيانات. لهذا عادةً نقوم بطباعة جملة له مثلاً:

```
echo "Please enter a value for the variable:"
read VARIABLE
```

المخطط البياني بالأسفل يوضح كيفية سير سكربت يطلب من المستخدم بإدخال بيانات:

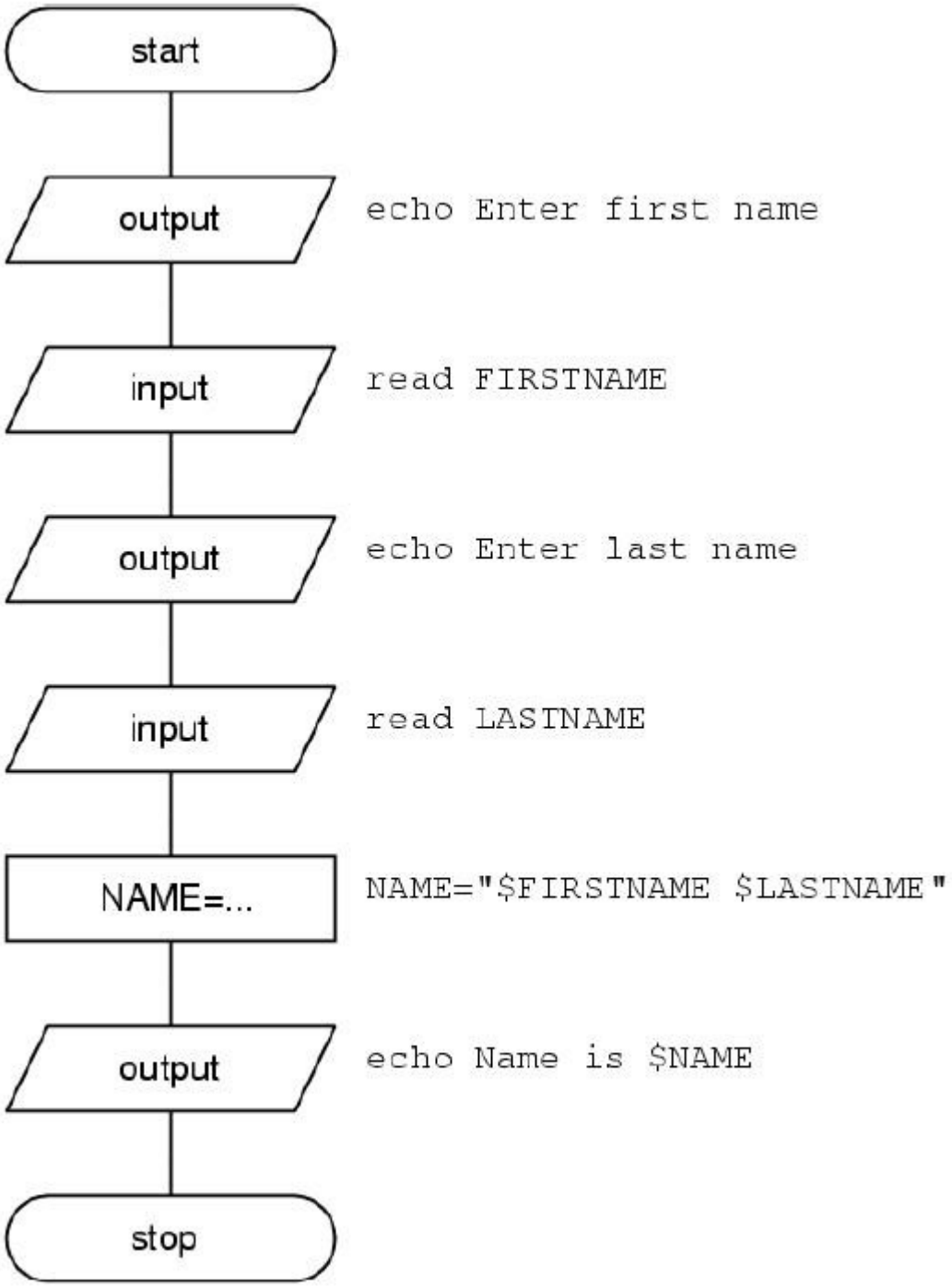


بالبداية يقوم السكربت بطباعة جملة معينة من خلال الأمر echo لإخبار المستخدم إن عليه إدخال بيانات معينة، ومن ثم يتوقف السكربت الى حين يقوم المستخدم بإدخال البيانات والضغط على Enter وبعد ذلك حسب المخطط الإنسيابي يقوم بطباعة هذه المدخلات التي تم تخزينها داخل المتغيير VARIABLE على الشاشة مرة أخرى بوساطة نفس الأمر echo.

تمرين: قم بكتابة سكربت يطلب من المستخدم إدخال إسمه الأول والأخير وبعد ذلك يقوم بالترحيب به بإستخدام إسمه الكامل.

كيفية إستخدام المتغيرات في السكربتات:

في هذا الجزء سنقوم بالتعرف على كيفية إستعمال المتغييرات داخل السكربت الذي سنقوم بكتابته. المخطط الإنسيابي الذي بالأسفل يوضح كيف يمكن إعطاء متغيير قيمة معينة.



```
#!/bin/bash
#B!n@ry 2nd Script
echo "Please enter your first name : "
read FIRSTNAME
echo "Please enter your last name : "
read LASTNAME
NAME="$FIRST $LAST"
echo Your Name is: $NAME
exit 0
```

المثال الذي بالأعلى يقوم بقراءة الأسم الأول ويضعه في المتغيير FIRSTNAME ومن ثم يخبرك بإدخال الإسم الثاني ويضعه في المتغيير LASTNAME، وبعد ذلك يقوم بإعطاء المتغيير الذي إسمه NAME قيمة كل من المتغيير FIRSTNAME و LASTNAME ويقوم بطباعتها لك. الآن لو نظرنا الى هذا السطر:
NAME=$FIRSTNAME $LASTNAME
حيث يوضح هذا السطر كيف أستطعنا من دمج القيمة التي في متغييرين في متغيير واحد هنا هو NAME. أيضاً ما أستفدنا من هذا المثال الذي بالأعلى هو إنه لتعيين قيمة الى متغيير معيين كل ما عليك فعله هو:

```
VARIABLE=Value
```

هذه في حالة التعيين. أما في حالة كنت تريد إسترجاع هذه القيمة فيمكنك إسترجاع هذه القيمة من خلال وضع علامة $ قبل المتغيير. مثلاً:

```
$VARIABLE
```

بعض المرات يفضل تعيين قيمة إفتراضية للمتغيير خاصة عندما يكون السكربت يتطلب إدخال قيمة معيينة. حيث هناك إحتمال المستخدم يدخل معلومة غلط أو قيم غلط تسبب في خروج السكربت عن تنفيذ المطلوب بصورة صحيحة. مثلاً:

```
NAME=${FIRSTNAME:=BINARY}
```

الآن لو حصل أي خطأ في إدخال قيمة المتغيير NAME فإنه سيأخذ القيمة الإفتراضية والتي هي هنا: BINARY.

```
#!/bin/bash
#B!n@ry 2nd Script
echo "Please enter your first name : "
read FIRSTNAME
echo "Please enter your last name : "
read LASTNAME
NAME="$FIRST $LAST"
echo Your Name is: $NAME
NAME=${FIRSTNAME:=BINARY}
echo $NAME
exit 0
```

كيفية إستعمال Command Substitution في السكربتات:
في هذا الجزء سنرى كيف يمكننا أن نستعمل نواتج الأوامر Command Substitution في السكربتات وكيفية الإستفادة من هذه الخاصية. لنأخذ مثلاً التالي:

```
#!/bin/bash
#B!n@ry 3rd Script
echo "Today is: `date +%m/%d/%Y`"
exit 0
```

حيث هذا الأمر يقوم بإرجاع قيمة التاريخ الحالي. أهم نقطة تنتبه لها، هي إن الأمر date والخيارات الخاصة به +m/%d/%Y% يجب أن توضع في ما يسمى backticks وهي الرمز ` . الآن بدل من طباعة الناتج الخاص بالأمر date مباشرة على الشاشة ممكن تخزينه في متغيير ومن ثم عرضه على الشاشة، ليصبح السكربت كالتالي:

```
#!/bin/bash
#B!n@ry 4th Script
TODAY=`date +%m/%d/%Y`
echo "Today is: $TODAY"
exit 0
```

في هذه الحالة تم تخزين ناتج الأمر date في متغيير إسمه TODAY ومن ثم قمنا بعرضه من بواسطة الأمر echo. نقطة ظرورية الإنتباه لها، وهي يجب أن لا يكون هناك فراغ بين المتغيير وعلامة الـ = والقيمة، أي لا يوجد فراغات بينهم نهائياً.

تمرين: قم بكتابة سكربت تطبع لك إسم المستخدم الذي تستخدمه في تلك اللحظة والمجلد الذي تعمل بداخله Current Working Directory. لحل هذا السؤال قم بإستعمال الأمر whoami والأمر pwd.

```
#!/bin/bash
#B!n@ry 6th Script
USERNAME=`whoami`
WORKINGDIR=`pwd`
echo "You are currently logged in with user: "
echo $USERNAME
echo "Your currently working directory is: "
echo $WORKINGDIR
exit 0
```

كيفية إستعمال المعادلات الرياضية في السكربتات:
في هذا الجزء سنقوم بتوضيح كيفية يمكن إستعمال العمليات الرياضية وكيفية الإستفادة منها داخل السكربتات.
عادةً تستعمل السكربتات القيم المخزنة في المتغيرات للعمليات الحسابية/الرياضية. لكن الـ BASH ليس مجهز بالكثير من الإمكانيات الحسابية/الرياضية ليقوم بهذه العمليات ولهذا يعتمد في تنفيذ هذه العمليات الحسابية الى أوامر خارجية مثل: expr وغيرها. الإمكنيات الوحيدة التي يستطيع عملها الـ BASH على مستوى العمليات الحسابية هي:
1- التعامل مع الأرقام الصحيحة Integers فقط.
2- جميع القيم يتم حجز 64 بت لها. ولهذا لن تتجاوز هذه القيم سالب 2 قوة 63 الى موجب 2 قوة 63-1.
لهذا عند الحاجة للتعامل مع الأرقم العشرية ستضطر الى إستعمال الأمر bc.

طرق إستعمال الـ BASH في المعادلات الحسابية:
- إستعمال الأمر الخارجي expr:

```
A=`expr $B + 10`
```

بما إنه الشل سكربت يسمح لنا بإستعمال الأوامر الخارجية إذن نستطيع ان نستعمل هذه الطريقة. نقطة مهمة هي السكربتات التي تستعمل الأوامر الخارجية ستكون أبطأ من تلك التي تستعمل الأوامر المبنية في الشل نفسه.

- إستعمال الأمر المبني في الشل let:

```
let A="$B + 10"
```

يمكننا الأمر let المبني في الـ BASH من إجراء العمليات الحسابية.

- إجراء العمليات الحسابية من خلال وضعها داخل أقواس مغلقة، هناك طريقتان:

```
A=$((B + 10))
```

أو

```
A=$[B + 10]
```

حيث يمكن للشل BASH أن يقوم بترجمة ما بين الأقواس وبالتالي حساب القيمة النهائية.

- إستعمال الأمر المبني في الشل والذي هو declare:

```
declare -i A
declare -i B
A=B+10
```

حيث قامت الأمر declare -i بتعريف كل من A و B على إنهم أرقام من نوع Integer أي رقم صحيح. قيامك بتعريف المتغييرات بهذه الطريقة سيهرف الشل على طريقة حسابهم بصورة آلية automatically ولا حاجة للمبرمج بوضع الإشارة $ قبل المتغيير حين يريد إستعماله.

ملاحظة مهمة:: عند إستعمالك للأمر expr فإنها تتعرف على العمليات الحسابية التالية فقط:
- الجمع +
- الطرح -
- القسمة /
- الضرب *
- النسبة المؤية % أو باقي القسمة

أما إن كنت تريد أن تستعمل العمليات الأخرى فلا ينفع إستعمالها مع الأمر expr وإنما يمكن ذلك بواسطة باقي الأوامر الذي ذكرت سابقاً. أيضاً يفضل أن تنفذ العمليات الحسابية هذه من خلال أحد الطرق المذكورة، وليس

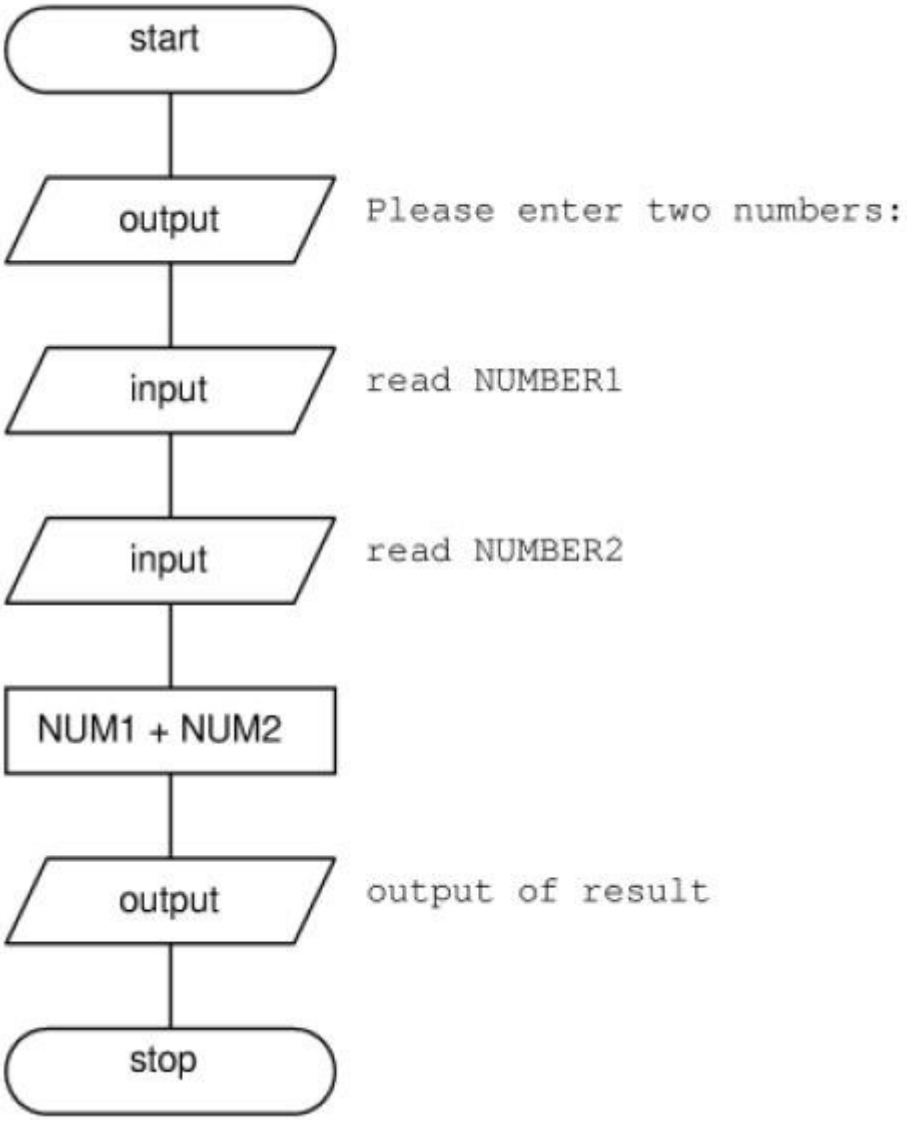
جميعها لأنها ستشتت طريقة تفكيرك وتعاملك مع العمليات الحسابية، وستجد نفسك في كل مرة تائه في مشكلة وضع المعادلة نفسها. وأنصح هنا بإستعمال الأمر declare حيث سيسهل عليك الكثير.

==========

للحصول على قائمة بكافة العمليات أذهب الى man bash

==========

تمرين: أنظر الى المخطط الإنسيابي التالي:



1- أكتب شل سكربت يعكس عمل المخطط أعلاه.
2- قم بالتعديل على السكربت لكي تستطيع تطبيق جميع العمليات الحسابية مثلا: الطرح، القسمة، والضرب.
3- ماذا سيحصل لو قام المستخدم بإدخال كلمة بدلاً من رقم ؟ وماذا سيحصل في حالة المستخدم مجرد ضغط على المفتاح Enter من دون إدخال معلومات ؟
4- قارن الحل الذي عملته مع الحل في آخر الكتاب.

==========

نهاية الجزء الأول من الكتاب

==========