1 Chamadas a scripts. Exportar variables

1.1 Sumario

1.2 Introdución

- Antes de comezar cos scripts que creen o esqueleto das carpetas do noso dominio, imos explicar brevemente como funciona a execución de scripts en Linux e a chamada entre eles:
    - Calquera proceso en Linux ten asignado un número de proceso.
    - Cada proceso en Linux non comparte nada con outros procesos e ten as súas propias variables.
    - As variables poden pasarse de procesos pai a procesos fillos con **export**
    - Cando se chama a un script, dependendo como se chame pódese crear un novo proceso fillo ou non.
    - A consola é un proceso, que ten as súas variables e o seu número de proceso correspondente.
    - **sh script.sh**: sh é un intérprete de comandos que abre un novo proceso fillo para executar o script.
    - **source script.sh**: *source* indica que se colla o contido do ficheiro *script.sh* e o execute como se as liñas se teclearan dende o proceso que chama ao script. Non se crea un proceso fillo, seguimos no mesmo proceso pai, só que o contido de *script.sh* foi traído ao proceso no que estamos.
    - **. script.sh**: é o mesmo que *source script.sh*
    - **PATH**: variable de contorna que indica os directorios nos que buscar executables.
    - Cando se inicia unha terminal ábrese un proceso que ten un id. este pódese consultar con **echo $$**

- Imos agora con varios exemplos de todo o anterior:

Abrimos un terminal e imos crear en */root/scripts* que vai conter os scripts

```
#Comezamos creando un directorio en dserver00 no que ter organizados tódolos scripts.
mkdir scripts
cd scripts
```

1.3 Os procesos e o contorno de traballo onde se executa un script

SCRIPT: pai.sh

```
#!/bin/bash

echo Son o script pai
echo Teño o proceso número: $$

CIDADE=Santiago

echo Como pai defino a variable Cidade, con valor $CIDADE
```

- **Probas iniciais:**

```
#Comprobamos que numero de proceso ten o terminal no que imos lanzar os scripts.
root@dserver00:~/scripts# echo $$
  3000

#Comprobamos en que ruta estamos.
root@dserver00:~/scripts# pwd
  /root/scripts

#Comprobamos que o ficheiro pai.sh existe e non ten permisos de execución
root@dserver00:~/scripts# ls pai.sh -l
  -rw-r--r-- 1 root root 233 Mai 10 13:39 pai.sh
```

1.3.1 Chamar a un script co inteprete sh

```
root@dserver00:~/scripts# sh pai.sh
  Son o script pai
  Teño o proceso número: 3830
  Como pai defino a variable Cidade, con valor Santiago

#Observar como o número do proceso do terminal é 3000 e o nº de proceso mentres se executaba o script era 3830
#Falamos en pasado porque ese proceso xa non existe unha vez que rematou o script.

#Miramos no terminal que valor ten a variable CIDADE que está dentro do script e que se acaba de executar
root@dserver00:~/scripts# echo $CIDADE
root@dserver00:~/scripts#

#Non ten valor no proceso da terminal. Non existe esa variable no terminal,
#pois unha vez que finalizou o script matouse o seu proceso asociado e
#as variables que nese proceso se definiran.
```

1.3.2 Executar un script con source ou con "."

- Cando se chama a un script con source ou "." é como se as liñas dese script se copiaran ao proceso actual, isto é, neste caso, na terminal:

```
#Antes de nada volvemos mirar no terminal que valor ten a variable CIDADE que está dentro do script
root@dserver00:~/scripts# echo $CIDADE
  root@dserver00:~/scripts#
#Polo explicado antes, non existe esa variable.

#Revisamos de novo o número de proceso do terminal, que era 3000
root@dserver00:~/scripts# echo $$
  3000

root@dserver00:~/scripts# . pai.sh
  Son o script pai
  Teño o proceso número: 3000
  Como pai defino a variable Cidade, con valor Santiago

# Observar agora que non se creou un novo proceso para executar o script, pois dentro do proceso 3000
# o que se fixo foi traer o contido do ficheiro e executar liña a liña dese ficheiro dentro do
# proceso 3000 (O terminal)

# Por tanto, que pasa agora coa variable cidade no terminal? no proceso 3000?
root@dserver00:~/scripts# echo $CIDADE
  Santiago

#Pois que agora a variable CIDADE existe no proceso 3000 (a terminal) e ten valor.
```

1.3.3 Executar o script chamándoo directamente polo seu nome. A variable de contorno PATH

```
#Para iso o script ten que ser executable. Por agora non o é.
root@dserver00:~/scripts# ls pai.sh -l
  -rw-r--r-- 1 root root 233 Mai 10 13:39 pai.sh
```

```
#Dámoslle permiso de execución ao script.
root@dserver00:~/scripts# chmod +x pai.sh
root@dserver00:~/scripts# ls pai.sh -l
  -rwxr-xr-x 1 root root 233 Mai 10 13:45 pai.sh

 #Chamamos ao script dende dentro do propio directorio que o contén. Pero dinos que non existe.
root@dserver00:~/scripts# pai.sh
  -bash: pai.sh: command not found

# Se miramos a variable PATH, vemos onde se buscan os ficheiros executables
# se non se lles indica unha ruta como no caso anterior
# A variable PATH é unha variable do sistema que herdan tódolos procesos fillos, entre eles a consola actual
root@dserver00:~/scripts# echo $PATH
  /usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin

# Antes de indicar como poderíamos modificar esa variable imos ver que pasa se lle poñemos a ruta ao script.
# Ruta absoluta:


# Poderíamos engadir á variable PATH
# Ruta absoluta ao script
root@dserver00:~/scripts# /root/scripts/pai.sh
  Son o script pai
  Teño o proceso número: 3852
  Como pai defino a variable Cidade, con valor Santiago

# Ruta relativa: dende o directorio actual no que estamos.
# Lembrar que o directorio no que estamos sempre se denomina . (non confundir co . de source).
root@dserver00:~/scripts# ./pai.sh
  Son o script pai
  Teño o proceso número: 3853
  Como pai defino a variable Cidade, con valor Santiago

# En ámbolos dous casos o script execútase pois puxemos unha ruta para chegar até el.
# Observar como se abriron novos procesos para executar o script en cada chamada.
# No proceso 3000 (a terminal) seguimos tendo a variable CIDADE da execución anterior con source ou .


#A variable PATH pódese modificar para a sesión actual do terminal (proceso 3000) como segue
PATH=$PATH:./ # Para engadir ao PATH actual o directorio no esteamos en cada momento
# ou
PATH=$PATH:/root/scripts # Ruta absoluta a onde se atopa os scripts que desexamos executar.
# Executamos agora o script simplemente chamándoo polo seu nome dende calquera sitio
root@dserver00:~/scripts# pai.sh
  Son o script pai
  Teño o proceso número: 3856
  Como pai defino a variable Cidade, con valor Santiago
# Outro novo proceso.

# Tan pronto se peche o terminal actual PATH volverá ao seu valor orixinal, perderanse os cambios realizados,
# pois eses cambios pertencen ao proceso 3000.
# Se desexamos facer permanentes eses valores podemos:
## Modificar a variable PATH en /etc/profile -> Afectaría a todo o sistema (Tódolos usuarios).
## Engadir PATH=$PATH:<directorios que se desexan> no fichiero .bashrc da carpeta persoal do usuario que
## desexa ter una PATH personalizado e que non afecte ao sistema nin a outros usuarios.
```

1.4 Chamadas entre procesos

1.4.1 Chamadas aos scripts dende o terminal

- Imos chamar a dous scripts. O segundo usa unha variable que define o primeiro, pero imos chamalos de forma independente dende a consola.

- Imos saír do terminal e volver entrar, para comezar de novo e que *PATH* non teñan en conta os cambios anteriores, nin exista a variable *CIDADE*.

- Creamos un novo script en */root/scripts* e situámonos nese directorio.

SCRIPT: fillo.sh

```
#!/bin/bash

echo Son o script fillo
echo Teño o proceso número: $$
echo Meu pai define a variable Cidade, con valor $CIDADE
```

1.4.1.1 Experimentación con sh

```
# O novo terminal ten o número de proceso 4000
root@dserver00:~/scripts# echo $$
  4000

#Chamamos ao proceso pai e execútase dentro dun novo proceso,
#que se destrúe ao rematar o script, e por tanto as variables que se usaran.
root@dserver00:~/scripts# sh pai.sh
  Son o script pai
  Teño o proceso número: 4023
  Como pai defino a variable Cidade, con valor Santiago

# Chamamos ao proceso fillo e execútase dentro dun novo proceso que se destrúe ao saír.
# Non pode coller o valor da variable CIDADE porque esa variable era do proceso anterior (4023) e
# destruíuse cando rematou o script/anterior.
root@dserver00:~/scripts# sh fillo.sh
  Son o script fillo
  Teño o proceso número: 4024
  Como pai defino a variable Cidade, con valor
```

1.4.1.2 Experimentación con source ou "."

```
root@dserver00:~/scripts# . pai.sh
  Son o script pai
  Teño o proceso número: 4000
  Como pai defino a variable Cidade, con valor Santiago
# Neste caso o contido de pai.sh executouse liña a liña dentro do proceso 4000 (a terminal).

# Chamamos ao proceso fillo.sh co intérprete sh.
# Pero vemos que se crea un novo proceso (4027) que non sabe cales son as variables
# que hai no proceso 4000. Por iso non sabe o valor da variable CIDADE
root@dserver00:~/scripts# sh fillo.sh
  Son o script fillo
  Teño o proceso número: 4027
  Meu pai define a variable Cidade, con valor

# Chamamos ao proceso fillo.sh dende o terminal con source ou .
# Co cal, estamos executando liña a liña dentro do proceso 4000.
# E como antes fixemos o mesmo con pai.sh,
# agora a variable CIDADE ten senso dentro do proceso 4000 (terminal)
root@dserver00:~/scripts# . fillo.sh
  Son o script fillo
  Teño o proceso número: 4000
  Meu pai define a variable Cidade, con valor Santiago.
```

1.5 Chamar a un script dende outro script: Export

- Para non interferir nesta última experimentación que imos facer, volvemos pechar o terminal e volvemos entrar.
- Para que a variable teña CIDADE, definida dentro de *pai.sh*, teña senso dentro de fillo.sh imos facelo de 2 maneiras:

1.5.1 Chamar dende o fillo ao pai

- Primeiro imos chamar dende *fillo.sh* ao script *pai.sh*

1.5.1.1 Chamar de fillo a pai: usando o intérprete sh

- Modificamos o contido dos script fillo.sh como segue:

SCRIPT: fillo.sh

```
#!/bin/bash

sh pai.sh
echo =======================================

echo Son o script fillo
echo Teño o proceso número: $$
echo Meu pai define a variable Cidade, con valor $CIDADE
```

- A liña 3 tamén podería ser **./pai.sh** se pai.sh tivera permisos de execución. Ou **pai.sh** se ademais dos permisos estivera o directorio no PATH

- Experimentación

```
# Dende o terminal chamamos a fillo.sh
# Créase un novo proceso para ese script. (4051)
# O script fillo.sh chama a script pai.sh que define a variable CIDADE pero que a destrúe
# tan pronto como remata a súa execución e devolve o control a fillo.sh que non sabe nada desa variable.
# Observar polos números de proceso que se crea antes o fillo cao pai.
root@dserver00:~/scripts# sh fillo.sh
  Son o script pai
  Teño o proceso número: 4050
  Como pai defino a variable Cidade, con valor Santiago
  Chamando a fillo.sh
  =======================================
  Son o script fillo
  Teño o proceso número: 4051
  Meu pai define a variable Cidade
```

1.5.1.2 Chamar de fillo a pai: usando source ou "."

- Modificamos o contido dos script fillo.sh como segue:

SCRIPT: fillo.sh

```
#!/bin/bash

. ./pai.sh  #é o mesmo que source ./pai.sh
echo =======================================

echo Son o script fillo
echo Teño o proceso número: $$
echo Meu pai define a variable Cidade, con valor $CIDADE
```

- A liña 3 tamén podería ser **. pai.sh**, pero cando se chamase ao script, no cando de **sh fillo.sh** deberamos chamalo como **. fillo.sh** (ou source fillo.sh). Neste caso o contorno de execución sería o do terminal, executaríase todo co mesmo número de proceso.

- Experimentación

```
# Dende o terminal chamamos a fillo.sh
# Créase un novo proceso para ese script.
# O script fillo.sh executa as liñas de pai.sh como se se as tivera el tecleadas no propio script, pois
# pai.sh é chamado con source ou "."
# Observar en que número de proceso se executan os dous scripts.
root@dserver00:~/scripts# sh fillo.sh
  Son o script pai
  Teño o proceso número: 4060
  Como pai defino a variable Cidade, con valor Santiago
  Chamando a fillo.sh
  =======================================
```

```
  Son o script fillo
  Teño o proceso número: 4060
  Meu pai define a variable Cidade, con valor Santiago
```

1.5.2 Chamar dende o pai ao fillo: export

- Veremos tamén dúas formas
- No segundo caso exportaremos unha variable do pai ao fillo.

1.5.2.1 Chamar de pai a fillo: usando o intérprete sh

- Modificamos o contido dos script **fillo.sh** como segue (Eliminamos a liña que chamaba ao script pai.sh):

SCRIPT: fillo.sh

```
#!/bin/bash

echo =======================================
echo Son o script fillo
echo Teño o proceso número: $$
echo Meu pai define a variable Cidade, con valor $CIDADE
```

- Modificamos o contido do script **pai.sh** como segue:

```
#!/bin/bash

echo Son o script pai
echo Teño o proceso número: $$

CIDADE=Santiago

echo Como pai defino a variable Cidade, con valor $CIDADE

echo Chamando a fillo.sh
sh fillo.sh
```

- Experimentación

```
# Dende o terminal chamamos a pai.sh
# Créase un novo proceso para ese script. (4070)
# O script pai.sh chama a script fillo.sh e crease un novo proceso (4071),
# Pero a variable CIDADE é so do contorno do pai, non dos fillos.
# Observar polos números de proceso cal se crea antes.
root@dserver00:~/scripts# sh pai.sh
  Son o script pai
  Teño o proceso número: 4070
  Como pai defino a variable Cidade, con valor Santiago
  Chamando a fillo.sh
  =======================================
  Son o script fillo
  Teño o proceso número: 4071
  Meu pai define a variable Cidade
```

1.5.2.2 Chamar de pai a fillo: exportando variables

- **Finalmente:**
- **Exportar unha variable do pai aos fillos: "export"**
- Modificamos o contido do script **pai.sh** como segue:

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

```
echo Son o script pai
echo Teño o proceso número: $$

CIDADE=Santiago
export CIDADE # As 2 liñas poderían quedar nunha soa: export CIDADE=Santiago

echo Como pai defino a variable Cidade, con valor $CIDADE

echo Chamando a fillo.sh
sh fillo.sh
```

- Experimentación

```
# Dende o terminal chamamos a pai.sh
# Créase un novo proceso para ese script. (4080)
# O script pai.sh chama a script fillo.sh e crease un novo proceso (4081),
# Pero antes de chamalo exportou a variable CIDADE, co cal esta vai estar dispoñible para
# tódolos scripts que se chamen dende o pai, aínda que os fillos se executen noutro proceso distinto.
# Observar polos números de proceso cal se crea antes, e como o proceso fillo ten acceso á variable CIDADE
root@dserver00:~/scripts# sh pai.sh
  Son o script pai
  Teño o proceso número: 4080
  Como pai defino a variable Cidade, con valor Santiago
  Chamando a fillo.sh
  ========================================
  Son o script fillo
  Teño o proceso número: 4081
  Meu pai define a variable Cidade, con valor Santiago
```

-- Antonio de Andrés Lema e Carlos Carrión Álvarez