Simply put, the shell is a program that takes your commands from the keyboard and gives them to the operating system to perform. In the old days, it was the only user interface available on a Unix computer.

Có nhiều loại shell, bash shell là enhanced version của sh shell nên cú pháp viết lệnh giống nhau. C shell thì khác bash shell và sh shell, c shell có cú pháp viết lệnh khác.

Khi ta su sang một user khác cũng có nghĩa là ta tạo ra một second shell từ first shell

[root@localhost ~]# echo $$

3655

[root@localhost ~]# bash

[root@localhost ~]# echo $$

3865

[root@localhost ~]# exit

exit

[root@localhost ~]# echo $$

3655

The first bash shell is 3655

The second bash shell is 3865

The second bash shell is created from the first bash shell using bash command.

* PID: Process Id
* PPID: Parent Process Id (the one which launched this PID)
* TGID: Thread Group Id

[root@localhost ~]# ps -p $$ -o "pid ppid cmd"

PID PPID CMD

3655 3654 bash

[root@localhost ~]# bash

[root@localhost ~]# ps -p $$ -o "pid ppid cmd"

PID PPID CMD

3972 3655 bash

-o format

User-defined format. format is a single argument in the form of a blank-separated or comma-separated list, which offers a way to specify individual output columns

[ian@atticf20 ~]$ **ps -p $$ -o "pid ppid cmd"**

PID PPID CMD

3175 2457 bash

[ian@atticf20 ~]$ **bash**

[ian@atticf20 ~]$ **ps -p $$ -o "pid ppid cmd"**

PID PPID CMD

4325 3175 bash

[ian@atticf20 ~]$ **VAR1=var1**

[ian@atticf20 ~]$ **VAR2=var2**

[ian@atticf20 ~]$ **export VAR2**

[ian@atticf20 ~]$ **export VAR3=var3**

[ian@atticf20 ~]$ **echo $VAR1 $VAR2 $VAR3**

var1 var2 var3

[ian@atticf20 ~]$ **echo $VAR1 $VAR2 $VAR3 $SHELL**

var1 var2 var3 /bin/bash

[ian@atticf20 ~]$ **ksh**

$ **ps -p $$ -o "pid ppid cmd"**

PID PPID CMD

4427 4325 ksh

$ **export VAR4=var4**

$ **echo $VAR1 $VAR2 $VAR3 $VAR4 $SHELL**

var2 var3 var4 /bin/bash

$ **exit**

[ian@atticf20 ~]$ **echo $VAR1 $VAR2 $VAR3 $VAR4 $SHELL**

var1 var2 var3 /bin/bash

[ian@atticf20 ~]$ **ps -p $$ -o "pid ppid cmd"**

PID PPID CMD

4325 3175 bash

[ian@atticf20 ~]$ **exit**

exit

[ian@atticf20 ~]$ **ps -p $$ -o "pid ppid cmd"**

PID PPID CMD

3175 2457 bash

[ian@atticf20 ~]$ **echo $VAR1 $VAR2 $VAR3 $VAR4 $SHELL**

/bin/bash

[ian@echidna ~]$ **ps -p $$ -o "pid ppid cmd"**

PID PPID CMD

2559 2558 -bash

[ian@echidna ~]$ **bash**

[ian@echidna ~]$ **ps -p $$ -o "pid ppid cmd"**

PID PPID CMD

2811 2559 bash

[ian@echidna ~]$ **VAR1=var1**

[ian@echidna ~]$ **VAR2=var2**

[ian@echidna ~]$ **export VAR2**

[ian@echidna ~]$ **export VAR3=var3**

[ian@echidna ~]$ **echo $VAR1 $VAR2 $VAR3**

var1 var2 var3

[ian@echidna ~]$ **echo $VAR1 $VAR2 $VAR3 $SHELL**

var1 var2 var3 /bin/bash

[ian@echidna ~]$ **ksh**

$ **ps -p $$ -o "pid ppid cmd"**

PID PPID CMD

2840 2811 ksh

$ **export VAR4=var4**

$ **echo $VAR1 $VAR2 $VAR3 $VAR4 $SHELL**

var2 var3 var4 /bin/bash

$ **exit**

[ian@echidna ~]$ **echo $VAR1 $VAR2 $VAR3 $VAR4 $SHELL**

var1 var2 var3 /bin/bash

[ian@echidna ~]$ **ps -p $$ -o "pid ppid cmd"**

PID PPID CMD

2811 2559 bash

[ian@echidna ~]$ **exit**

exit

[ian@echidna ~]$ **ps -p $$ -o "pid ppid cmd"**

PID PPID CMD

2559 2558 -bash

[ian@echidna ~]$ **echo $VAR1 $VAR2 $VAR3 $VAR4 $SHELL**

/bin/bash

**Notes:**

1. At the start of this sequence, the bash shell had PID 3175.
2. The second bash shell has PID 4325, and its parent is PID 3175, the original bash shell.
3. We created VAR1, VAR2, and VAR3 in the second bash shell, but only exported VAR2 and VAR3.
4. In the Korn shell, we created VAR4. The echo command displayed values only for VAR2, VAR3, and VAR4, confirming that VAR1 was not exported. Were you surprised to see that the value of the SHELL variable had not changed, even though the prompt had changed? You cannot always rely on SHELL to tell you what shell you are running under, but the ps command does tell you the actual command. Note that ps puts a hyphen (-) in front of the first bash shell to indicate that this is the *login shell*.
5. Back in the second bash shell, we can see VAR1, VAR2, and VAR3.
6. And finally, when we return to the original shell, none of our new variables still exist.

When you create a shell variable, you will often want to export it to the environment so it will be available to other processes that you start from this shell. Variables that you export are **not** available to a parent shell. You use the export command to export a variable name.