

# Shell Script

23/02/2018

Jing Maoqiang

# What can shell script do?

- Put many commands into a script, so you can avoid to forget the commands you have to type.
- When you need to batch files, shell script is a efficiency and good choice.
- With shell script, you can document every step and its purpose.

## Weakness of shell script

- Very low efficiency for calculation.

# Common Linux orders and some Linux knowledge

- A complete file name should be specified in the path , the path has a relative path and absolute path, if you don't specify the path of a file, it will be considered in the current path. E.g.
  - /afs/ihep.ac.cn/users/j/jingmq(“/” is root path)
  - ~/work (“~” can represent home path)
  - ../work (a relative path)
- “..” represents the upper directory.
- “.” represents current path.
- pwd, passwd, ls, cd, mkdir, rm, cp, mv, du, history, sed ………

# How can we put commands into shell script

- Primary commands you need to write into shell script:
  - “#!/usr/bin/env bash”/”#!/bin/bash” /” #!/bin/sh”(this has to be put in the beginning of shell script, every line beginning with “#” won’t be executed except this one. **By the way, I don’t understand what’s differences between these commands.**)
  - Or you can put nothing special in the beginning, but the commands can only be executed line by line and can’t loop, judge and so on.
  - If no special instructions, we consider we have put “#!/usr/bin/env bash”/”#!/bin/bash”/” #!/bin/sh” in the beginning afterwards.
- Next step you have to do is putting the commands into shell script and make some comments with “#”.
- Finally, you have to use command “chmod 700 \*\*\*” to change permissions of your shell script and use “bash” or “./” to run it.

# Some suggestions about writing shell script

- Use “\$” to claim a variable and take its value, and after claiming you can use double quotes to add the variable into a file name you want to create. E.g.
  - `mv Phieta_2.1750.sh Phieta_"$i".sh`
- Use apostrophe to take a variable's value in sed command. E.g.
  - `sed -i s/'2.1750/'$i'/g jobjpsi.foot`
- Use “touch A” to create a file named A and use “echo “\*\*\*” >> A” to add something into A. E.g.
  - `SIM_NAME=$JOB_NAME"_0"$INPUT".txt"`
  - `echo "//*****job options for EvtGen*****" >> $SIM_NAME`
- Use “ls -d” you can get directory's name without showing contents in it.

# Some suggestions about writing shell script

- Use “\” to convert some special character in shell script.
  - `sed -i "s/+ \ /-0,/ /g" /besfs/users/jingmq/workarea665p01/run/Phieta/ISR/$i/${j}`
- How to define a function to generate a random number:
  - `function rand(){`
  - `min=$1`
  - `max=$((($2-$min+1))`
  - `num=$(date +%s%N)`
  - `echo $((($num%$max+$min))`
  - `}`
  - Use “`dominantrnd=$(rand 0 90)`” to generate a random number between 0 and 90.
- Use “`ls -d`” you can get directory’s name without showing contents in it.

# Some suggestions about writing shell script

- How to do accumulation:
  - `let "j = $j + 1"`
- How to multiply and divide:
  - `NormRnd=`echo "scale=8; 0.35 * ${IntRnd[$num]} / $sum " | bc -l``
  - Use “scale” to control precision of the output result.
  - Must note: `NormRnd= `echo "scale=8; 0.35 * ${IntRnd[$sum]} / $sum " | bc -l`` is wrong, "NormRnd=" can't be followed by blankspace

**Thank you !**