# 语法：

$\*表示所有这些参数都被双引号引住。若一个脚本接收两个参数，$\*等于$1$2

$@表示所有这些参数都分别被双引号引住，若一个脚本接收到两个参数，$@等价于$1$2

$#表示提供给脚本的参数号

运行test.sh 1 2 3后

$\*为"1 2 3"（一起被引号包住）

$@为"1" "2" "3"（分别被包住）

$#为3（参数数量）

# bash\_case

|  |
| --- |
| #!/bin/sh  #$#:传递给程序的总的参数数目  #$n:表示第几个参数，$1 表示第一个参数，$2 表示第二个参数 ...  #$0:当前程序的名称  #$\*:传递给程序的所有参数组成的字符串  #$@:以"参数1" "参数2" ... 形式保存所有参数  case "$1" in  "a" | "A" )  echo A  ;;  "b" | "B" )  echo B  ;;  "c" | "C" )  echo C  ;;  \*)  echo no parameters  ;;  esac |

# bash\_param

|  |
| --- |
| #!/bin/sh  #$#:传递给程序的总的参数数目  #$n:表示第几个参数，$1 表示第一个参数，$2 表示第二个参数 ...  #$0:当前程序的名称  #$\*:传递给程序的所有参数组成的字符串  #$@:以"参数1" "参数2" ... 形式保存所有参数  #compare strings  if [ "$1" = "abcde" ];then  echo $1 match "abcde"  elif [ "$1" = "none" ];then  echo $1 match "none"  else  echo $1 not match "abcde" or "none"  fi |

# 循环读cpu状态：

|  |
| --- |
| while true; do  echo " "  cat /proc/stat  sleep 1  done |

# 循环读寄存器值

|  |
| --- |
| while true; do  echo " "  busybox devmem 0xe0168000 32  busybox devmem 0xe01b0054 32  sleep 0.5  done |

# preview

|  |
| --- |
| #!/system/bin/sh  while [ true ]; do  echo "Goto take a picture..."  am start -n com.android.camera2/com.android.camera.CameraActivity  sleep 5  input keyevent HOME  sleep 1  echo "done"  done |

# take\_picture

|  |
| --- |
| #!/system/bin/sh  echo "Goto take a picture..."  am start -n com.android.camera2/com.android.camera.CameraActivity  sleep 5  input keyevent ENTER  sleep 3  input keyevent ENTER  sleep 3  input keyevent HOME  sleep 1  echo "done"  exit 0 |

# take\_video

|  |
| --- |
| #!/system/bin/sh  echo "Goto take a video..."  am start -n com.android.camera2/com.android.camera.CameraActivity  sleep 5  input keyevent DPAD\_DOWN  sleep 3  input keyevent ENTER  sleep 10  input keyevent ENTER  sleep 3  input keyevent HOME  sleep 1  echo "done"  exit 0 |

# bash\_freq

|  |
| --- |
| #!/system/bin/sh  case "$1" in  "1" )  freq=696000  echo 1 : ${freq}  echo "performance" > /sys/devices/system/cpu/cpu0/cpufreq/scaling\_governor  echo ${freq} > /sys/devices/system/cpu/cpu0/cpufreq/scaling\_max\_freq  ;;  "2" )  freq=1008000  echo 2 : ${freq}  echo "performance" > /sys/devices/system/cpu/cpu0/cpufreq/scaling\_governor  echo ${freq} > /sys/devices/system/cpu/cpu0/cpufreq/scaling\_max\_freq  ;;  "3" )  freq=1260000  echo 3 : ${freq}  echo "performance" > /sys/devices/system/cpu/cpu0/cpufreq/scaling\_governor  echo ${freq} > /sys/devices/system/cpu/cpu0/cpufreq/scaling\_max\_freq  ;;  "4")  freq=1404000  echo 4 : ${freq}  echo "performance" > /sys/devices/system/cpu/cpu0/cpufreq/scaling\_governor  echo ${freq} > /sys/devices/system/cpu/cpu0/cpufreq/scaling\_max\_freq  ;;  "5")  freq=1536000  echo 5 : ${freq}  echo "performance" > /sys/devices/system/cpu/cpu0/cpufreq/scaling\_governor  echo ${freq} > /sys/devices/system/cpu/cpu0/cpufreq/scaling\_max\_freq  ;;  \*)  echo no parameters  echo 1 : 696000  echo 2 : 1008000  echo 3 : 1260000  echo 4 : 1404000  echo 5 : 1536000  ;;  esac |

|  |
| --- |
|  |

|  |
| --- |
|  |

# end