

# Level 1

---

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

```
echo "欢迎来到石头剪刀布游戏!"
echo "请在1.石头2.剪刀3.布三项中进行选择"
read choice
computer_choice=$((RANDOM % 3 + 1))
case $choice in
  1) # 用户选择了石头
    case $computer_choice in
      1)
        echo "电脑选择了石头，平局!"
        ;;
      2)
        echo "电脑选择了剪刀，恭喜您赢了!"
        ;;
      3)
        echo "电脑选择了布，很遗憾，您输了!"
        ;;
    esac
    ;;
  2) # 用户选择了剪刀
    case $computer_choice in
      1)
        echo "电脑选择了石头，很遗憾。您输了!"
        ;;
      2)
        echo "电脑选择了剪刀，平局!"
        ;;
      3)
        echo "电脑选择了布，恭喜您赢了!"
        ;;
    esac
    ;;
  3) # 用户选择了布
    case $computer_choice in
      1)
        echo "电脑选择了石头，恭喜您赢了!"
        ;;
      2)
        echo "电脑选择了剪刀，很遗憾你输了!"
        ;;
      3)
        echo "电脑选择了布，平局!"
        ;;
    esac
  esac
```

```

    ;;
esac

*) # 用户输入了除了1,2,3三个数字之外的内容
echo "无效的选择, 请重新输入!"

;;
esac

```

```

[root@centos diurnal]# ./rock_scissors_paper.sh
欢迎来到石头剪刀布游戏！
请在1.石头2.剪刀3.布三项中进行选择
1
电脑选择了布，很遗憾，您输了！
[root@centos diurnal]# ./rock_scissors_paper.sh
欢迎来到石头剪刀布游戏！
请在1.石头2.剪刀3.布三项中进行选择
2
电脑选择了布，恭喜您赢了！
[root@centos diurnal]# ./rock_scissors_paper.sh
欢迎来到石头剪刀布游戏！
请在1.石头2.剪刀3.布三项中进行选择
3
电脑选择了石头，恭喜您赢了！
[root@centos diurnal]#

```

## Level 2

```

#!/bin/bash
# 其中的注释大部分为我自己哪里不懂哪里查出来的，我尽量去理解这些命令的意图并学习
# 检查内存使用情况
check_memory(){
    total_memory=$((free -m | awk '/Mem:/ {print $2}'))
    used_memory=$((free -m | awk '/Mem:/ {print $3}'))
    # 这两行中，free -m 命令用于查询系统内存的使用情况，再利用管道符将结果输入到右边；awk 是一个文本处理工具，用于逐行分析和处理文本文件，/Mem:/ 是一种模式，用于匹配含“Mem:”的行；print $2 是一个行为，用于打印该行的第二个字段
    memory_percentage=$((used_memory * 100 / total_memory))
    # 由于Shell中的算术运算只能处理整数，因此使用$(())语法来计算，只使用一个()会导致被解释成字符串中的赋值操作
    threshold=80 # 设置阈值为80
    if ((memory_percentage >= threshold));then # 比较大小为算术运算，因此使用(( ))
        echo "内存使用超过阈值:${memory_percentage}% " # 百分号的作用是使结果带上百分号，如果超过阈值则会输出此结果
    else
        echo "您的内存使用合理，当前使用占比:${memory_percentage}%"
    fi
}

# 检查磁盘使用情况
check_disk(){
    threshold=80 # 设置阈值为80

    while read -r line; do # -r选项用于禁止对反斜杠的转义，确保一些特殊符号不会被意外解释，通常while read结构用于读取文本文件或命令输出，允许逐行读取记录并对其进行处理(例如分段或转换)，同时整个结构会从df -h中逐行读取内容，存储到line变量中
        disks=$(echo "$line" | awk '{print $1}') # 使用awk命令提取$line中的第一个字段(磁盘名称)，将结果保存在disk变量中
        usage=$(echo "$line" | awk '{print $5}') # 使用awk命令提取$line中的第五个字段，并通过sed命令删除百分号，将结果保存在usage变量中

        if ((usage >= threshold));then
            echo "磁盘 ${disk} 使用超过阈值:${usage}%"
        else
            echo "您的磁盘 ${disk} 使用合理,当前使用占比:${usage}%"
        fi
        done < <(df -h | awk 'NR>1') # df -h命令的输出为"列出文件系统的磁盘使用情况"，作为输入传递给while循环进行处理，awk 'NR>1'用于跳过标题行，从第二行开始处理
    }

    check_memory
    check_disk
}

```

```
[root@centos system]# systemctl status monitor.service
● monitor.service - System monitor
   Loaded: loaded (/etc/systemd/system/monitor.service; enabled; vendor preset: disabled)
   Active: inactive (dead) since 五 2023-11-03 23:55:48 CST; 1min 0s ago
   Main PID: 22963 (code=exited, status=0/SUCCESS)

11月 03 23:55:48 centos systemd[1]: Started System monitor.
11月 03 23:55:48 centos monitor.sh[22963]: 您的内存使用合理, 当前使用占比:64%
11月 03 23:55:48 centos monitor.sh[22963]: 您的磁盘 /dev/sda3 使用合理,当前使用占比:33%
11月 03 23:55:48 centos monitor.sh[22963]: 您的磁盘 devtmpfs 使用合理,当前使用占比:0%
11月 03 23:55:48 centos monitor.sh[22963]: 您的磁盘 tmpfs 使用合理,当前使用占比:0%
11月 03 23:55:48 centos monitor.sh[22963]: 您的磁盘 tmpfs 使用合理,当前使用占比:2%
11月 03 23:55:48 centos monitor.sh[22963]: 您的磁盘 tmpfs 使用合理,当前使用占比:0%
11月 03 23:55:48 centos monitor.sh[22963]: 您的磁盘 /dev/sda1 使用合理,当前使用占比:50%
11月 03 23:55:48 centos monitor.sh[22963]: 您的磁盘 tmpfs 使用合理,当前使用占比:1%
[root@centos system]# systemctl status monitor.timer
● monitor.timer - System monitor timer
   Loaded: loaded (/etc/systemd/system/monitor.timer; enabled; vendor preset: disabled)
   Active: active (waiting) since 五 2023-11-03 17:53:11 CST; 6h ago

11月 03 17:53:11 centos systemd[1]: Started System monitor timer.
[root@centos system]#
```