第3天服务器信息清洗

一、 subprocess 执行本机命令

> `subprocess` 的 `getoutput()` 方法可以接收一个字符串的参数，这个参数会被认为是当前操作系统的命令去执行，并返回字符串类型的命令执行结果，假设命令出错，会返回相应的错误信息。

比如我们执行一条命令，来获取的厂商机器信息 目标是得到这些信息

> - 厂商 就是 `Manufacturer` 对应的值 比如 Del
> - 服务器型号(名字) 就是 `Product Name` 对应的值 比如 Del R710
> - 服务器序列号 就是 `Serial Number` 对应的值

```
[root@qfedu.com ~]# rpm -q dmidecode
dmidecode-3.1-2.el7.x86_64
[root@qfedu.com ~]# dmidecode -q -t 1 2>/dev/null
System Information
    Manufacturer: Alibaba Cloud
    Product Name: Alibaba Cloud ECS
    Version: pc-i440fx-2.1
    Serial Number: 0f7e3d86-7742-4612-9f93-
e3a9e4754157
    UUID: 0f7e3d86-7742-4612-9f93-e3a9e4754157
    Wake-up Type: Power Switch
    SKU Number: Not Specified
    Family: Not Specified

```

在 python 中可以这样做

```
In [1]: import subprocess

In [2]: prod_info = "dmidecode -q -t 1 2>/dev/null"

In [3]: prod_info
Out[3]: 'dmidecode -q -t 1 2>/dev/null'

In [4]: subprocess.getoutput(prod_info)
Out[4]: 'System Information\n\tManufacturer: Alibaba Cloud\n\tProduct Name: Alibaba Cloud ECS\n\tVersion: pc-i440fx-2.1\n\tSerial Number: 0f7e3d86-7742-4612-9f93-e3a9e4754157\n\tUUID: 0f7e3d86-7742-4612-9f93-e3a9e4754157\n\tWake-up Type: Power Switch\n\tSKU Number: Not Specified\n\tFamily: Not Specified\n'

In [5]:
```

可以看到输出结果是一个整体的字符串，和 shell 中输出的少有不同，就是这里每一行后面都有一个换行符 '\n' 那我们要想对每一行进行处理，可以使用 `split('\n')` 进行分割，当然我们这里使用另一个方法 `splitlines()`, 它默认使用的分隔符就是换行符

```
In [5]: ret = subprocess.getoutput(prod_info)

In [6]: ret.splitlines()
Out[6]:
['System Information',
 '\tManufacturer: Alibaba Cloud',
 '\tProduct Name: Alibaba Cloud ECS',
 '\tVersion: pc-i440fx-2.1',
 '\tSerial Number: 0f7e3d86-7742-4612-9f93-e3a9e4754157',
```

```
 '\tUUID: 0f7e3d86-7742-4612-9f93-e3a9e4754157',
 '\tWake-up Type: Power Switch',
 '\tSKU Number: Not Specified',
 '\tFamily: Not Specified']

In [7]:
```

那接着我们即可以循环列表中的每个元素（也就是每行），在循环中处理每行内容，得到我们想要的数据

```
In [7]: for line in ret.splitlines():
   ...:     print(line)
   ...:
System Information
    Manufacturer: Alibaba Cloud
    Product Name: Alibaba Cloud ECS
    Version: pc-i440fx-2.1
    Serial Number: 0f7e3d86-7742-4612-9f93-
e3a9e4754157
    UUID: 0f7e3d86-7742-4612-9f93-e3a9e4754157
    Wake-up Type: Power Switch
    SKU Number: Not Specified
    Family: Not Specified

In [8]: for line in ret.splitlines():
   ...:     if 'Manufacturer:' in line:
   ...:         print(line)
   ...:
    Manufacturer: Alibaba Cloud
```

可以看到我们拿到了我们需要的第一个数据，并且可以进行进一步的处理，比如转换成一个字典，其他可以如法炮制。

```
In [12]: prod_dic = {}
    ...: for line in ret.splitlines():
```

```
    ...:     k = ''
    ...:     line = line.strip()
    ...:     print(line)
    ...:     if ': ' in line:
    ...:         k, v = line.split(': ')
    ...:         print(k)
    ...:     if k == 'Manufacturer':
    ...:         prod_dic[k.lower()] = v
    ...:     elif k == 'Product Name':
    ...:         prod_dic[k.lower()] = v
    ...:     elif k == 'Serial Number':
    ...:         prod_dic[k.lower()] = v
    ...:

In [13]: prod_dic
Out[13]:
{'serial_number': '0f7e3d86-7742-4612-9f93-e3a9e4754157',
 'manufacturer': 'Alibaba Cloud',
 'product_name': 'Alibaba Cloud ECS'}
```


和我们的目标越来接近了，但你会发现有问题

1. 多个判断语句导致代码臃肿
2. 并且 if 语句中存在重复的代码

那继续优化，思路是可以提前定义一个映射的字典

```
#!/usr/bin/env python3
import subprocess

prod_info = 'dmidecode -q -t 1 2>/dev/null'

# 执行系统命令并返回结果
ret = subprocess.getoutput(prod_info)
```

```
prod_dic = {}

# 定义映射字典
map_dic = {
    "Manufacturer": "manufacturer",
    "Product Name": "pod_name",
    "Serial Number": "sn"
}

for line in ret.splitlines():
    line = line.strip()
    try:  # 异常处理语句
        k, v  = line.split(": ")
        if k in map_dic:
            # k = map_dic.get(k)
            prod_dic[map_dic.get(k)] = v
    except ValueError as e:
        print(e)
# print('....>>>')
print(prod_dic)
# 输出信息
{'manufacturer': 'VMware, Inc.', 'pod_name':
'VMware7,1', 'sn': 'VMware-56 4d 2b 4b 91 1e 48 15-
5b d2 73 9c ec 98 da 22'}

```


二、 获取服务器的硬件基础信息

1. 基础信息

```
# 主机名
cmd_machine = 'uname -n'

# 内核版本
cmd_kernel = 'uname -r'

# 操作系统
cmd_os_version = "cat /etc/redhat-release
```

2. 厂家和产品信息

```
[root@qfedu.com]# dmidecode -q -t 1 2>/dev/null
System Information
    Manufacturer: Alibaba Cloud                    # 厂商
    Product Name: Alibaba Cloud ECS          #  机器型号
    Version: pc-i440fx-2.1
    Serial Number: 0f7e3d86-7742-4612-9f93-e3a9e4754157
    UUID: 0f7e3d86-7742-4612-9f93-e3a9e4754157
    Wake-up Type: Power Switch
    SKU Number: Not Specified
    Family: Not Specified
```


3. CPU 信息

3.1 查看物理CPU型号

```
grep 'model name' /proc/cpuinfo | uniq
```

```
In [1]: import subprocess

```

```
In [2]: cmd_cpu_name = "grep 'model name' /proc/cpuinfo | uniq"

In [3]: subprocess.getoutput(cmd_cpu_name)
Out[3]: 'model name\t: Intel(R) Xeon(R) Platinum 8163 CPU @ 2.50GHz'

In [4]: cpu_name = subprocess.getoutput(cmd_cpu_name).split(": ")[1]

In [5]: cpu_name
Out[5]: 'Intel(R) Xeon(R) Platinum 8163 CPU @ 2.50GHz'

In [6]: cpu = {"cpu_name": cpu_name}

In [7]: cpu
Out[7]: {'cpu_name': 'Intel(R) Xeon(R) Platinum 8163 CPU @ 2.50GHz'}

In [8]:
```


3.2 查看物理CPU颗数

```
grep 'physical id' /proc/cpuinfo | sort -u | wc -l
```

```
In [8]: cmd_cpu_pyc = "grep 'physical id' /proc/cpuinfo | sort -u | wc -l"

In [9]: subprocess.getoutput(cmd_cpu_pyc)
Out[9]: '1'

In [10]: cpu["pyc"] = int(subprocess.getoutput(cmd_cpu_pyc))

In [11]: cpu
Out[11]: {'cpu_name': 'Intel(R) Xeon(R) Platinum 8163 CPU @ 2.50GHz', 'cpu_pyc': 1}

In [12]:
```

3.3 查看每颗物理 CPU 的核心数

```
grep 'cpu cores' /proc/cpuinfo | uniq   # 每颗 CPU 的核心数，不是总核心数
```

```
In [13]: subprocess.getoutput("grep 'cpu cores' /proc/cpuinfo | uniq")
Out[13]: 'cpu cores\t: 1'

In [14]: cpu_cores_each = subprocess.getoutput("grep 'cpu cores' /proc/cpuinfo | uniq")

In [15]: cpu_cores_each = int(cpu_cores_each.split(": ")[1])

In [16]: cpu_cores_each
Out[16]: 1

In [17]: cpu["cores_each"] = cpu_cores_each
```

```
In [18]: cpu
Out[18]:
{'cpu_name': 'Intel(R) Xeon(R) Platinum 8163 CPU @ 2.50GHz',
 'cpu_num': 1,
 'cpu_cores_each': 1}

In [19]:
```

4. 内存信息

- 阿里云虚拟主机

```
[root@qfedu.com]# dmidecode -q -t 17 2>/dev/null
Memory Device
    Total Width: Unknown
    Data Width: Unknown
    Size: 4096 MB                # 容量
    Form Factor: DIMM
    Set: None
    Locator: DIMM 0              # 插槽号
    Bank Locator: Not Specified
    Type: RAM                    # 类型  物理的有 DDR3 DDR4
    Type Detail: Other
    Speed: Unknown               # 速率  物理的有 1333 等
    Manufacturer: Alibaba Cloud
    Serial Number: Not Specified
    Asset Tag: Not Specified
    Part Number: Not Specified
    Rank: Unknown
    Configured Clock Speed: Unknown
    Minimum Voltage: Unknown
```

```
    Maximum Voltage: Unknown
    Configured Voltage: Unknown
```

- 物理 R710 服务器

```
Memory Device
    Total Width: 72 bits
    Data Width: 64 bits
    Size: 8192 MB
    Form Factor: DIMM
    Set: 6
    Locator: DIMM_B2
    Bank Locator: Not Specified
    Type: DDR3
    Type Detail: Synchronous Registered (Buffered)
    Speed: 1333 MT/s
    Manufacturer: 00CE00B380CE
    Serial Number: 82B79F71
    Asset Tag: 02120363
    Part Number: M393B1K70DH0-YH9
    Rank: 2
Memory Device
    Total Width: 72 bits
    Data Width: 64 bits
    Size: 8192 MB
    Form Factor: DIMM
    Set: 6
    Locator: DIMM_B3
    Bank Locator: Not Specified
    Type: DDR3
    Type Detail: Synchronous Registered (Buffered)
    Speed: 1333 MT/s
    Manufacturer: 00CE00B380CE
    Serial Number: 32CDDE81
    Asset Tag: 02120361
```

```
    Part Number: M393B1K70CH0-YH9
    Rank: 2
Memory Device
    Total Width: 72 bits
    Data Width: 64 bits
    Size: No Module Installed
    Form Factor: DIMM
    Set: 4
    Locator: DIMM_B4
    Bank Locator: Not Specified
    Type: DDR3
    Type Detail: Synchronous
    Speed: Unknown
    Manufacturer:
    Serial Number:
    Asset Tag:
    Part Number:
    Rank: Unknown
Memory Device
    Total Width: 72 bits
    Data Width: 64 bits
    Size: 8192 MB
    Form Factor: DIMM
    Set: 5
    Locator: DIMM_B5
    Bank Locator: Not Specified
    Type: DDR3
    Type Detail: Synchronous Registered (Buffered)
    Speed: 1333 MT/s
    Manufacturer: 00CE04B380CE
    Serial Number: 85966B82
    Asset Tag: 02113621
    Part Number: M393B1K70DH0-YH9
    Rank: 2
Memory Device
```

```
    Total Width: 72 bits
    Data Width: 64 bits
    Size: 8192 MB
    Form Factor: DIMM
    Set: 6
    Locator: DIMM_B6
    Bank Locator: Not Specified
    Type: DDR3
    Type Detail: Synchronous Registered (Buffered)
    Speed: 1333 MT/s
    Manufacturer: 000000B380CE
    Serial Number: 00000000
    Asset Tag: 02121563
    Part Number:
    Rank: 2
```

作业

请使用以上信息，编写一个脚本，输出如下信息


```
{
    "base_info": {
        "host_name": "db_server",
        "kernel": "3.10.0-957.21.3.el7.x86_64",
        "os": "CentOS Linux release 7.6.1810 (Core)",
        'manufacturer': 'Alibaba Cloud',
        'pod_name': 'Alibaba Cloud ECS',
        'sn': '0f7e3d86-7742-4612-9f93-e3a9e4754157'
    },
    "cpu": {
        'name': 'Intel(R) Xeon(R) Platinum 8163 CPU @ 2.50GHz',
        'num': 1,
```

```
            'cores_each': 1
        },
        "mem": [
            {
                'capacity': '8192 MB',
                'slot': 'DIMM_A3',
                'model': 'DDR3',
                'speed': '1333 MT/s',
                'manufacturer': '00CE04B380CE',
                'sn': '8362A2F8'
            },
            {
                'capacity': 'No Module Installed',
                'slot': 'DIMM_A4',
                'model': 'DDR3',
                'speed': 'Unknown'
            }
            ............略............
        ]
    }
```


内存源数据使用上面 R710 的，映射字典使用下面这个

```
key_map = {
            'Size': 'capacity',
            'Locator': 'slot',
            'Type': 'model',
            'Speed': 'speed',
            'Manufacturer': 'manufacturer',
            'Serial Number': 'sn',
}

```

内存处理参考代码

```
def parse(data):
    key_map = {
        'Size': 'capacity',
        'Locator': 'slot',
        'Type': 'model',
        'Speed': 'speed',
        'Manufacturer': 'manufacturer',
        'Serial Number': 'sn'
    }


   info_mem = []
    # 首先把服务器上的所有插槽分开，并发到一个列表中
    # 这个语法叫列表生成式， 表示循环中的元素为真时候，将 mem 添加到列表中
    memory_list = [ mem for mem in data.split('Memory Device') if mem]


    for item in memory_list:
        # 把每个插槽的信息放到一个字典中
        single_slot = {}

        for line in item.splitlines():
            line = line.strip()
            if len(line.split(': ')) == 2:
                key, val = line.split(': ')
                if key in key_map:
                    # 获取到映射字典的 value 作为新字典的 key
                    single_slot[key_map[key]] = val
        # 含有插槽信息的字典:
```

```
30          # {'capacity': '8192 MB', 'slot': 'DIMM_A3',
    'model': 'DDR3', 'speed': '1333 MT/s',
    'manufacturer': '00CE04B380CE', 'sn': '8362A2F8'}
31
32          # 由于存在多个内存插槽，每个插槽的号码是不一样的
33          # 所以可以把当前内存的插槽号作为总体内存字典中的一个
    key，值就是当前含有插槽信息的字典
34          info_mem.append = single_slot
35      return info_mem
```