

浪潮信息云峦服务器操作系统KeyarchOS FAQ手册

浪潮电子信息产业股份有限公司

2023年6月

目录

[目录 I](#_Toc137795955)

[1 前言 4](#_Toc137795956)

[2 日志收集 4](#_Toc137795957)

[2.1 安装过程日志 4](#_Toc137795958)

[2.2 系统运行日志 5](#_Toc137795959)

[2.2.1 sosreport日志 5](#_Toc137795960)

[2.2.2 串口日志 5](#_Toc137795961)

[2.2.3 内存转储kdump 9](#_Toc137795962)

[2.2.4 mcelog 13](#_Toc137795963)

[2.2.5 dmesg 13](#_Toc137795964)

[2.2.6 message日志不打印 14](#_Toc137795965)

[3 安装失败 14](#_Toc137795966)

[3.1 Your BIOS-based system needs a special partition to boot from a gpt disk label. To continue, please create a 1Mib biosboot type partition. 14](#_Toc137795967)

[3.2 通过bmc挂载镜像的方式安装系统时，无法启动安装的图形界面，只能在文本模式下安装 16](#_Toc137795968)

[3.3 通过BMC挂载镜像安装KOS时，选择Install kos 5后，在启动日志打印中出现dracut-initqueue timeout报错信息 17](#_Toc137795969)

[3.4 安装KOS时在安装界面报错“Some packages from local repository have incorrect checksum” 17](#_Toc137795970)

[3.5 安装源配置出错，需要排查ISO sha256值 17](#_Toc137795971)

[3.6 使用UltraISO制作U盘启动盘，出现无法启动安装问题 18](#_Toc137795972)

[4 启动异常/宕机/假死/黑屏 19](#_Toc137795973)

[4.1 系统下运行yum remove libaio之后系统重启进入emergency mode 19](#_Toc137795974)

[4.2 系统根目录占满导致无法启动图形界面 20](#_Toc137795975)

[4.3 系统下存在两块相同的lvm格式磁盘，无法启动 20](#_Toc137795976)

[5 CPU部分 21](#_Toc137795977)

[5.1 cpuinfo中core id的排序乱序 21](#_Toc137795978)

[5.2 不同服务器上查看numa node的cpu编号可能不同 21](#_Toc137795979)

[5.3 Processor heated above trip temperature. Throttling enabled 22](#_Toc137795980)

[5.4 短时间内发生的cpu温度告警问题 22](#_Toc137795981)

[6 磁盘部分 23](#_Toc137795982)

[6.1 磁盘乱序 23](#_Toc137795983)

[6.2 fdisk -l 和 lsblk 显示磁盘大小不同 28](#_Toc137795984)

[6.3 如何在安装过程中清理磁盘 29](#_Toc137795985)

[7 网络部分 29](#_Toc137795986)

[7.1 avahi-daemon: Host name conflict, retrying with linux-2 29](#_Toc137795987)

[7.2 Activation via systemd failed for unit 'dbus-org.freedesktop.resolve1.service': Unit dbus-org.freedesktop.resolve1.service not found. 30](#_Toc137795988)

[8 内存部分 32](#_Toc137795989)

[8.1 Out of Memory: Killed process [PID] [process name] 32](#_Toc137795990)

[8.2 重启对比/proc/meminfo的内存容量发现内存大小有细微的变化 32](#_Toc137795991)

[9 显示部分 33](#_Toc137795992)

[9.1 journal: failed to get edid data: EDID length is too small 33](#_Toc137795993)

[9.2 failed to get edid: unable to get EDID for output 33](#_Toc137795994)

[10 其他部分 34](#_Toc137795995)

[10.1 BMC控制台大小写键盘混乱 34](#_Toc137795996)

[10.2 每次重启messages中的启动时间会快8小时，直到Journal启动后恢复正常时间 34](#_Toc137795997)

[10.3 如何在KOS系统中读取服务器SN信息 36](#_Toc137795998)

[10.4 如何使用系统镜像配置本地yum源 36](#_Toc137795999)

# 前言

本文档的适用对象是浪潮信息云峦服务器操作系统KeyarchOS（简称KOS）产品的POC测试、实施、售后维护人员。

本文档对KOS产品在POC测试、实施、售后阶段频繁遇到的问题进行了总结，建议相关技术人员提前阅读本文档，以便更高效的完成相关技术工作。

如需安装KOS操作系统，可从以下位置下载：

|  |  |
| --- | --- |
| 条目 | 下载地址 |
| 镜像位置 | https://zh.ieisystem.com/product/kos/9453.html |

# **日志收集**

## 安装过程日志

在安装过程中，使用ctrl+alt+F2切换到字符界面，将/tmp目录下所有文件打包，并通过挂载U盘拷出。

# tar -czvf install\_log.tar.gz /tmp

适用场景：系统可以正常启动到安装界面，在安装过程中遇到问题，但未宕机可响应键盘操作。

## 系统运行日志

### sosreport日志

sosreport 是一个从运行中的系统收集大量配置细节、系统信息和诊断信息的工具，它可以获取以下日志：

○ 操作系统版本、已安装的内核、已加载的内核模块、打开的文件列表、PCI 设备列表

○ 引导信息、文件系统、内存、主机名、已安装的RPM

○ 系统 IP、网络详细信息、系统路由

○ 挂载点、位于 /etc 文件夹中的所有配置文件，以及位于 /var 文件夹中的所有日志文件

○ 运行中的进程信息、进程树输出

**常用收集方法：**使用root用户执行 sosreport --batch或sos report --batch 命令，约1分钟后在/var/tmp/下生成一个sosreport.xxx.tar.xz的压缩包文件。

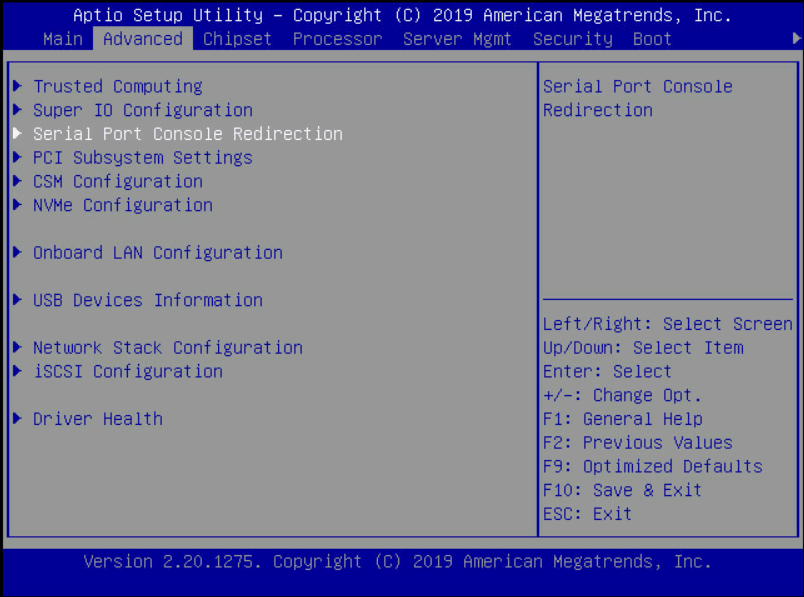
### 串口日志

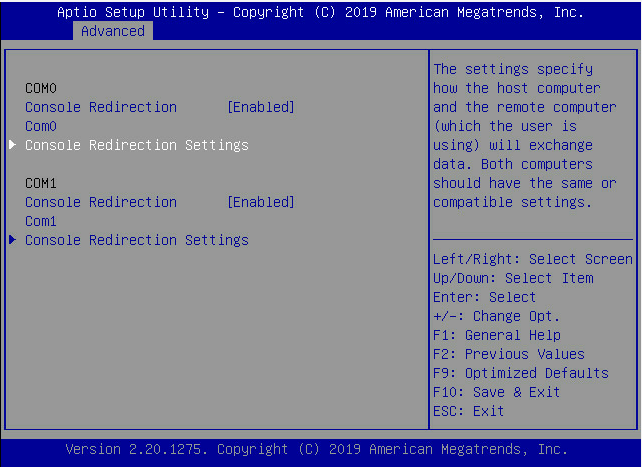
在服务器测试和系统日常使用过程中，偶尔会遇到系统崩溃之类的严重错误，此时操作系统会将部分重要信息输出至串口。连接物理串口获取这些信息十分不便，因此建议进行串口重定向配置，使得用户可以利用串口获取这些信息。

KOS串口重定向配置方法可参考如下（不同服务器BIOS配置界面可能不同）：

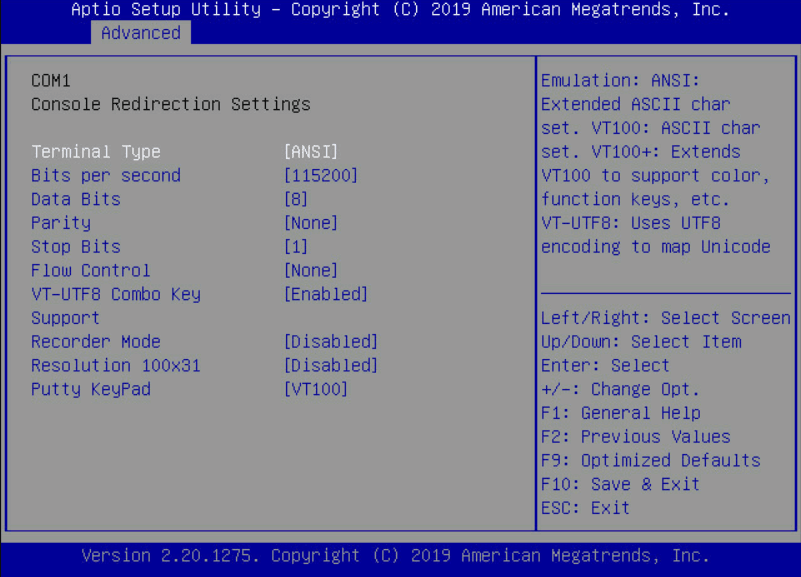
#### BIOS配置开启串口重定向

1、选项Advanced->Serial Port Console Redirection->Console Redirection中的COM端口设置为Enable





2、配置相关COM端口及波特率参数



#### KOS配置GRUB串口参数

1. 增加串口参数：

# grubby --args=”console=tty0 console=ttyS0,115200” --update-kernel=”/boot/vmlinuz-<对应内核>”

2、重启

# reboot

3、检查配置生效

# cat /proc/cmdline

#### 收集串口日志

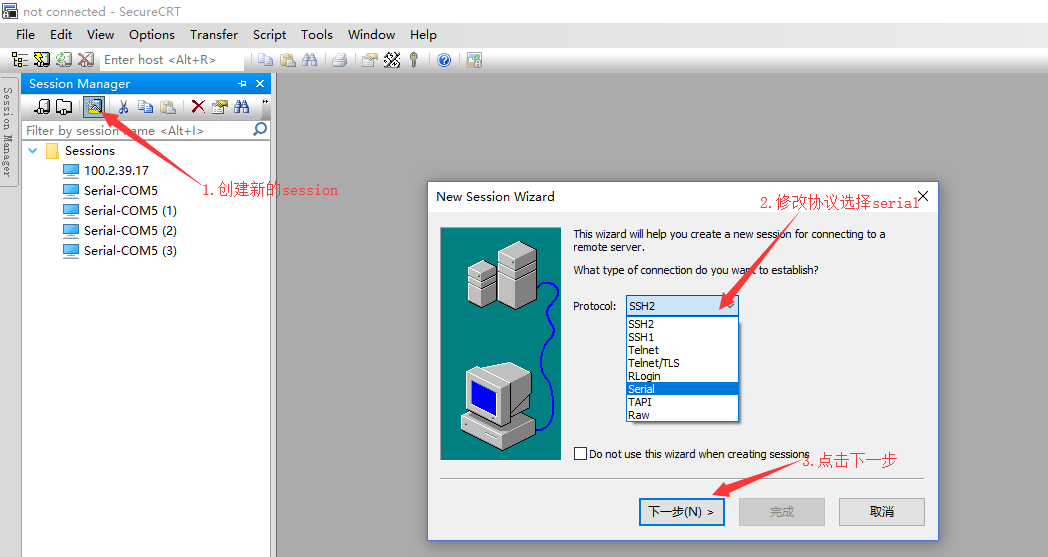
（1）方法一：

在任何一台可以和BMC连通的机器上，执行：

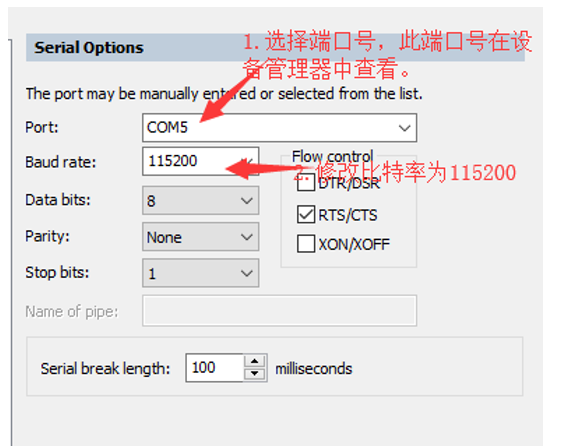
#ipmitool -I lanplus –H服务器BMC地址–U服务器BMC用户名–P服务器BMC密码sol activate > 本地文件

（2）方法二：

使用串口转接线连接笔记本和服务器BMC管理口，笔记本上使用远程连接工具，如SecureCRT，配置串口连接。



配置串口端口号和比特率，点击连接，即可在当前窗口看到服务器输出串口信息，将输出的日志保存到文件中。



### 内存转储kdump

当系统崩溃时，需要收集足够的信息来分析引起崩溃的原因，内存转储是一种在系统崩溃时将系统的内存信息以DUMP文件的形式保存下来的机制。

kdump是利用kexec实现的可靠的内核崩溃转储机制。在系统崩溃时使用kexec重新启动第二个内核，去捕获并转储内核崩溃信息。kdump利用kexec可以不经过 BIOS直接使用预留的内存启动转储内核，在系统崩溃后捕获保留第一内核崩溃信息以供调试分析，是用来分析系统崩溃、内核panic、死锁时常用的工具和服务。

#### 2.2.3.1 修改启动crashkernel参数

KOS系统默认自带kdump设置，对于ANCK和RHCK内核默认有不同的crashkernel策略，以下分别说明：

ANCK内核默认参数如下：

|  |  |
| --- | --- |
| ANCK | system memory:memory reserved by crashkernel |
| x86\_64 | 0G-2G:0M,2G-8G:192M,8G-:256M |
| aarch64 | 0G-2G:0M,2G-8G:192M,8G-:768M |

若需要手工修改的话，按如下方法操作：

grubby --args=”crashkernel=0M-2G:0M, 2G-8G:192M,8G-:\*\*\*M”

--update-kernel=”/boot/vmlinuz-<对应内核>”

重启后可通过kdumpctl showmem命令确认是否修改成功。

RHCK内核默认参数如下：

|  |  |
| --- | --- |
| RHCK | 参数 |
| x86\_64 | crashkernel=auto |
| aarch64 | crashkernel=auto |

若需要手工修改的话，按如下方法操作：

grubby --args=”crashkernel=\*\*\*M”--update-kernel=”/boot/vmlinuz-<对应内核>”

重启后可通过kdumpctl showmem命令确认是否修改成功。

对于RHCK内核，建议x86\_64架构下配置256M，aarch64架构下配置768M。

注：（1）crashkernel=auto是一种算法计算的内存预留，可能无法满足所有系统的需求（尤其是对于具有许多 IO 卡和加载内核驱动程序的配置），需要通过测试 kdump 确保 crashkernel=auto 保留的内存足以供目标机器使用。

（2）针对Ampere CPU crashkernel需要设置的到2G以上。

#### 2.2.3.2 配置转储目录

kdump转储目录在/etc/kdump.conf中配置，默认转储到/var/crash/下。

如果需要转储到其他目的地，例如转储到其他磁盘目录，需要在文件所在的存储设备前声明文件系统类型：

ext4 /dev/sda1

path=/usr/local/crash

转储到NFS设备：

net \*<nfs server>:</nfs/mount>\*

比如：net nfs.example.com:/export/vmcores

转储到可SSH访问的设备：

net \*<user>@<ssh server>\*

比如：net root@ssh.example.com

#### 2.2.3.3 配置触发条件

通过编辑/etc/sysctl.conf设置系统触发Kdump的条件：

# vim /etc/sysctl.conf

配置完成后，执行如下命令立即生效：

# sysctl -p

|  |  |
| --- | --- |
| kernel.sysrq=1/0 | #是否响应sysrq魔术键触发panic |
| kernel.unknown\_nmi\_panic=1/0 | #硬件NMI按钮被按下时是否触发panic |
| kernel.panic\_on\_unrecovered\_nmi=1/0 | #发生unrecovered的NMI时是否触发panic |
| vm.panic\_on\_oom=1/0 | #发生内存OOM时是否触发panic |
| kernel.panic\_on\_oops = 1/0 | #内核在发生oops错(非法内存访问或非法指令)时是否触发panic |
| kernel.softlockup\_panic = 1/0 | #内核在发生软锁(softlockup)时是否触发panic |
| kernel.panic\_on\_io\_nmi = 1/0 | #内核收到因I/O错误导致的NMI时是否触发panic |
| kernel.hung\_task\_panic = 1/0 | #内核在进程hung住时，是否触发panic |
| kernel.hung\_task\_timeout\_secs=120 | #进程hung住多少时间，触发panic |
| kernel.hardlockup\_panic = 1/0 | #发生hardlockup时是否触发panic |

注：以上系统内核参数，正常情况下按系统默认配置即可，在需要debug相关问题时，根据需要配置开启相关参数触发kdump定位问题。

#### 2.2.3.4 开启kdump服务

系统下的kdump服务，需要在系统下正常运行，并默认开机自启。

# systemctl restart kdump

# systemctl enable kdump

# systemctl status kdump

#### 2.2.3.5 重启检查服务生效

# reboot

# systemctl status kdump

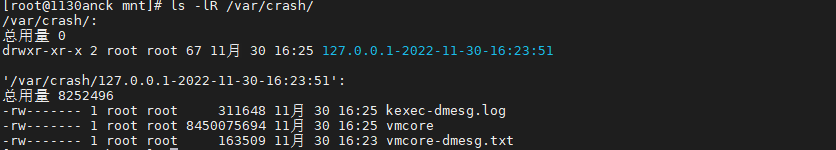
#### 2.2.3.6 验证触发kdump

测试kdump配置是否生效时，需要SysRq机制的支持才能使用，检查并确保sysrq开启：

# echo 1 > /proc/sys/kernel/sysrq

# echo c > /proc/sysrq-trigger

重启后在kdump.conf文件配置的转储目录下，存在vmcore文件即代表配置成功。



### mcelog

KOS中，默认mcelog服务不创建/var/log/mcelog文件，mcelog相关的内容会默认记录到messages文件中，若需要单独的mcelog文件，需要修改daemon服务的参数，加上“--logfile=/var/log/mcelog”，并重启mcelog服务。

**# systemctl stop mcelog**

**# vim /usr/lib/systemd/system/mcelog.service**

ExecStart=/usr/sbin/mcelog --ignorenodev --daemon --foreground --logfile=/var/log/mcelog

**# systemctl daemon-reload**

**# systemctl start mcelog**

**# systemctl status mcelog**

### dmesg

KOS及较新的Linux版本中，/var/log/dmesg都已经不再默认生成，因为/var/log/messages日志中已经包含了系统启动日志。

KOS中可以通过如下方法重新开启记录 /var/log/dmesg：

# vim /etc/systemd/system/dmesg.service

[Unit]

Description=Create /var/log/dmesg on boot

ConditionPathExists=/var/log/dmesg

[Service]

ExecStart=/usr/bin/dmesg

StandardOutput=[file:/var/log/dmesg](http://file/var/log/dmesg)

[Install]

WantedBy=multi-user.target

# touch /var/log/dmesg

# restorecon -v /var/log/dmesg

# systemctl enable dmesg

# systemctl start dmesg

### message日志不打印

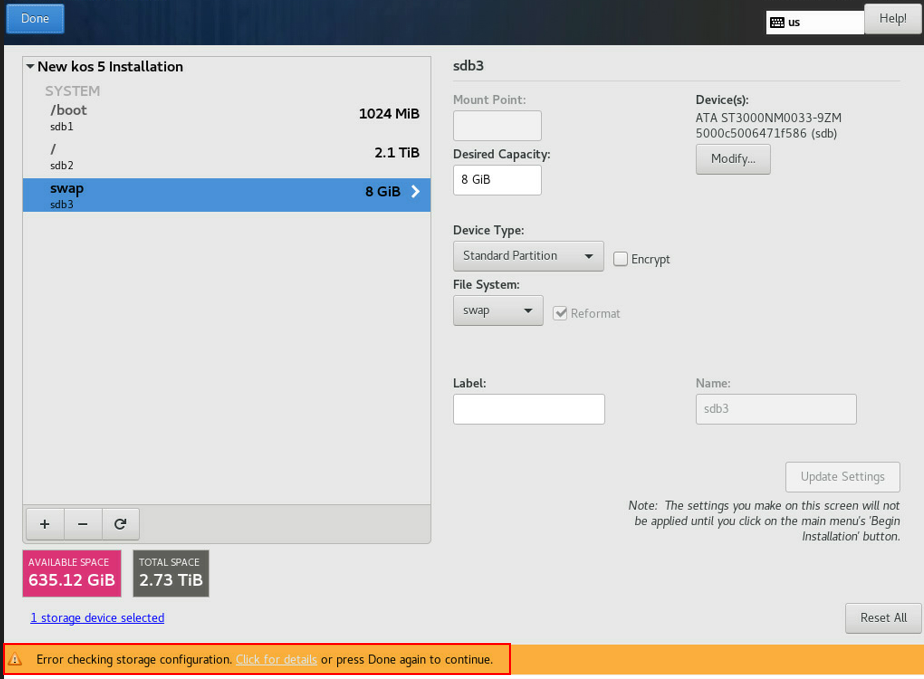
请确保rsyslog服务保持启动状态。

# 安装失败

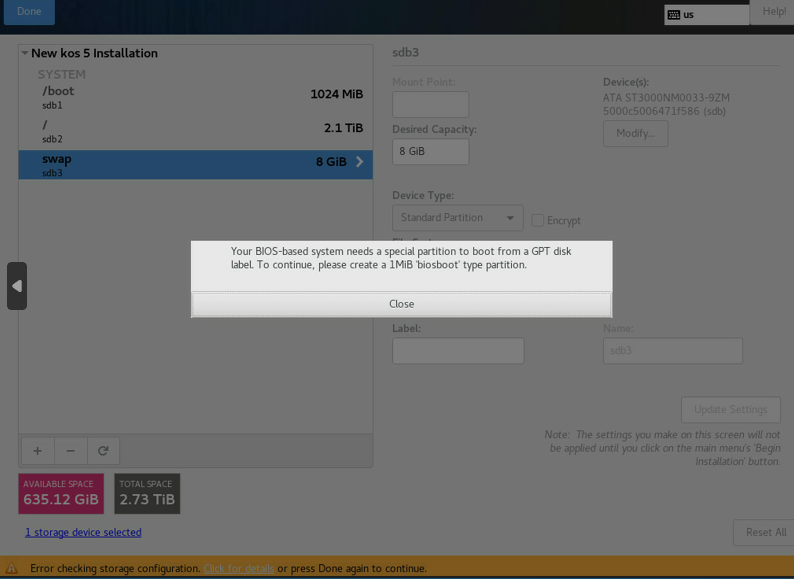
## Your BIOS-based system needs a special partition to boot from a gpt disk label. To continue, please create a 1Mib biosboot type partition.

【问题描述】

Legacy模式下使用超过2T空间的磁盘安装系统，手动分区后点击Done会提示如下报错：



点击Click for details，显示如下：



【原因说明】

在legacy模式安装系统时，使用gpt分区的磁盘，需要创建biosboot分区，用来识别大于2T的gpt分区磁盘。

【解决方案】

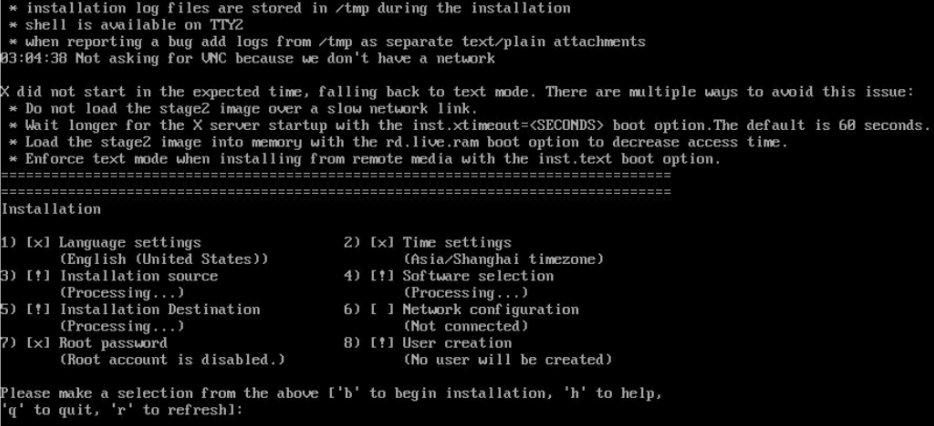
如果使用kickstart，增加part biosboot --fstype=biosboot --size=1；

如果使用图形界面安装，增加1MB大小的biosboot分区。

## 通过bmc挂载镜像的方式安装系统时，无法启动安装的图形界面，只能在文本模式下安装

【问题描述】

通过bmc挂载镜像的方式安装系统时，无法启动安装的图形界面，只能在文本模式下安装。



【原因说明】

由于通过bmc挂载镜像安装时，受网络环境的影响，加载Xserver比较慢，预期时间(默认60S)内未加载完成，所以切换成文本模式进行安装。

【解决方法】

根据提示在安装系统时添加inst.xtimeout=<SECONDS>（默认60s）grub参数来延长X Server加载时间。

## 通过BMC挂载镜像安装KOS时，选择Install kos 5后，在启动日志打印中出现dracut-initqueue timeout报错信息

该问题由于启动vmlinuz和initramfs文件系统时超时导致，导致超时的原因是由于网络波动造成img传输问题，从而导致报错，出现该问题后可检查网络环境或者通过ＵＳＢ本地安装。

## 安装KOS时在安装界面报错“Some packages from local repository have incorrect checksum”

出现该问题的原因可能由以下几种可能：

a. 如通过挂载ISO安装，请检查镜像哈希值或者由于网络传输问题导致，软件包哈希值不对。

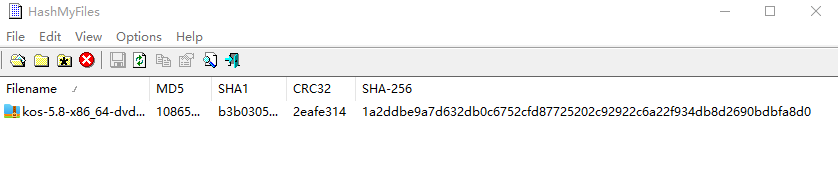
b.如通过U盘安装，请检查启动盘是否正确。

## 安装源配置出错，需要排查ISO sha256值

【原因说明】

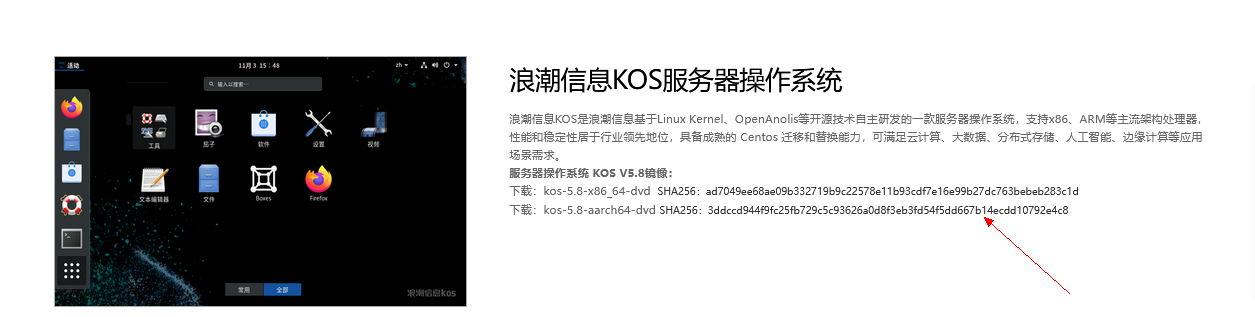
刻录U盘安装或挂载ISO镜像安装时界面显示repo错误或者日志显示软件包checksum错误，请检查ISO镜像是否完整。检查方法如下：

1.在windows系统下使用HashMyFiles.exe 工具生成iso镜像的sha256值：



2.查看官网下载链接中的sha256值：

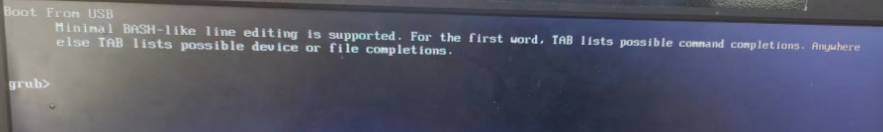
<https://zh.ieisystem.com/product/kos/9453.html>



3.对比HashMyFiles.exe 工具生成的sha256与官网下载链接中的sha256值：如果一致则iso是完整的；如果不一致iso镜像就是不完整的。

## 使用UltraISO制作U盘启动盘，出现无法启动安装问题

如图，使用UItraISO制作U盘启动盘后，无法启动安装：



请确认UItraISO工具的写入方式需为RAW 。



# 启动异常/宕机/假死/黑屏

## 系统下运行yum remove libaio之后系统重启进入emergency mode

【问题描述】

在系统下运行yum remove libaio之后系统重启进入emergency mode

【原因说明】

在执行yum remove libaio时可以看到会卸载58个依赖包，其中包括了lvm2的包，造成在系统重启的时候出现/home分区无法挂载问题。

【解决方案】

进入emergency之后使用yum install lvm2可以解决，但是由于卸载的其他包太多，所以建议重装系统。

## 系统根目录占满导致无法启动图形界面

【问题描述】

当出现无法登录图形界面，但可以远程连接机器的情况时，可通过以下步骤检查并确认是否根目录被占满导致无法启动图形界面：

1、系统启动时，去除“rhgb quiet”参数，看到系统服务启动失败；

2、重新启动系统，修改grub参数增加“3”，进入runlevel 3，系统可以正常启动，再通过init 5进入图形界面也可正常进入；

3、系统下发现根目录 / 占用率100%，有可能导致系统图形界面启动时的相关服务启动异常；迁移释放根目录空间后，重启进入系统正常。

【原因说明】

系统根目录空间占用100%，若默认使用图形界面登录，有可能导致系统图形界面启动时的相关服务启动异常，无法进入系统。

## 系统下存在两块相同的lvm格式磁盘，无法启动

【问题描述】

启动时报错：multiple LVM volume groups with the same name(kos)

【原因说明】

系统下 lvm vg id 不能重复

【解决方案】

挂载镜像，按3进入shell；

输入pvs -o +vg\_uuid 获取vg list；

使用vgrename  [VG UUID] vg-newname修改vg name。

修改后会破坏启动选项，该盘无法继续作为系统盘。

# **CPU部分**

## cpuinfo中core id的排序乱序

**【问题描述】**

通过“cat /proc/cpuinfo | grep 'core id'”抓取core id的排序乱序，不是按照core id的大小顺序排列的。

**【问题原因】**

这种乱序不是问题，在OS中是正常的，因为core id在OS中是一个逻辑值，和物理CPU没有对应关系，所以不能保证逻辑core id的值是顺序的。

**【解决办法】**

不是问题，忽略

## 不同服务器上查看numa node的cpu编号可能不同

**【问题描述】**

# numactl --hardware

available: 2 nodes (0-1)

node 0 cpus: 0 1 2 3 4 5 6 7 16 17 18 19 20 21 22 23

node 1 cpus: 8 9 10 11 12 13 14 15 24 25 26 27 28 29 30 31

 # numactl --hardware

available: 2 nodes (0-1)

node 0 cpus: 0 2 4 6 8 10 12 14 16 18 20 22 24 26 28 30

node 1 cpus: 1 3 5 7 9 11 13 15 17 19 21 23 25 27 29 31

不同服务器上执行numactl --hardware命令，查看numa node的cpu编号可能不同。

**【问题原因】**

内核无法设置某个CPU（不论是物理CPU还是逻辑CPU）的编号，内核只能通过ACPI接口从BIOS中获取CPU编号，内核根据BIOS提供的apicid来作为CPU的编号，具体每个Node上CPU编号是由BIOS和不同的CPU型号决定的。

**【解决办法】**

根据具体的CPU型号咨询Intel和BIOS。

## Processor heated above trip temperature. Throttling enabled

【问题描述】

在mcelog中看到的温度告警信息Processor 48 heated above trip temperature. Throttling enabled.

【原因】

Intel CPU有两个温度告警阈值：TM1和TM2，分别控制CPU节流和关闭系统。当到达TM1时，CPU会报告一个thermal event 给mcelog。

【解决方法】

检查散热相关硬件

## 短时间内发生的cpu温度告警问题

【问题描述】

系统日志中存在下面的报错：

Oct 7 15:34:26 localhost mcelog: Processor 48 heated above trip temperature. Throttling enabled.

...

Oct 7 15:34:26 localhost mcelog: Processor 48 below trip temperature. Throttling disabled

【原因】

短时间的告警不会显著降低应用程序的性能，并且通常在处理器变化的正常范围内。

这样的事件可能会导致操作系统发出机器检查警报，系统的功能并不受这些警报的影响，特别是在运行动态处理器压力应用程序时。

【解决方法】

在短时间内发生的cpu温度告警"heated above trip temperature"，在同1s内又报below trip temperature，可以忽略。

# **磁盘部分**

## 磁盘乱序

【问题描述】

操作系统重启或热插拔的场景中，可能会发现磁盘设备的sd[X]名称变化，比如由原来的sda变为sdc，或磁盘设备的名称sd[X]与磁盘设备的物理位置不是一一对应的，此种现象统称为磁盘乱序。

【原因说明】

操作系统每次重启，磁盘设备名称都有可能变化，这是因为操作系统中驱动加载的顺序，磁盘扫描顺序都不是固定的；Linux并没有固定的驱动加载顺序，磁盘设备的发现是一个并行和异步的过程，通常磁盘会按照被识别的顺序依次进行初始化，但是，盘符的命名（比如 sda）是按照驱动加载完成最先响应的顺序决定的。在Server重启、设备Reconfiguration等操作时有可能出现盘符变动，完全无法预测盘符与设备的物理槽位的对应关系。对于SCSI总线上的磁盘设备，扫描分配盘符的过程不仅取决于 SCSI INQUIRY, TEST UNIT READY等命令的响应，还受在此期间系统上运行的进程数量、udev工作者线程数量、连接到Server的其他外围设备数量以及初始化这些PCI设备所花费的CPU cycles的影响，也依赖于服务器上可用的CPU cores 数量、内存大小等，这些因素可能导致设备初始化所需时间的微小变化，因此可能会得到不同的“/dev/sdx”名称。

这个现象在Linux中是正常现象，这并不是一个缺陷，而是Linux就是这样设计的，设备名（盘符）的设计初衷并不是用来和磁盘的物理位置或顺序进行一一对应的，应使用设备或数据自身提供的唯一标识来区分设备。

磁盘乱序的问题，在所有Linux版本中都存在，KOS也不例外。

【解决方案】

1. **首选方案使用设备自标识符**

使用设备或分区/LVM卷/文件系统自身提供的标识**，**而不是使用/dev/sd[X]等由内核分配给每个设备使用的设备名称。建议使用磁盘的 /dev/disk/by-path (by-id或by-uuid等) 来标识永久磁盘名。

在grub引导行中，就使用了不会在系统重新引导时发生变化的标识符标识根设备：

root=/dev/mapper/vg1-root //LVM卷名称 root=UUID=f2dc94c1-6124-4368-b215-f04ab171e024 //文件系统的UUID root=PARTUUID=7d3bb988-02d0-4c6c-a864-14f4acfd0398 //分区UUID

同样，在/etc/fstab中，指定的分区名称也不依赖于sd[X]名称：

/dev/mapper/vg1-root / ext4 defaults 1 1

UUID=0c12f366-3dfa-4ea9-923c-5a57994ceef8 /boot ext4 defaults 1 2 **二、使用udev规则绑定盘符**

udev是用于创建和命名与系统中存在的设备对应的/dev设备节点的机制，udev使用sysfs提供的匹配信息和用户提供的规则来动态添加所需的设备节点。

对于磁盘的唯一标识名称，使用udev规则来对磁盘的SERIAL\_ID/WWID/PATH等设备的唯一索引做别名的方案，并不是对磁盘设备名进行修改，而是在kernel分配的磁盘设备名不变的基础上，添加一个别名，实际上磁盘乱序的现象仍然存在，只是从操作系统下提供了一个“不乱序的”设备名。

具体实现脚本举例：由于脚本是使用当前系统下的lsscsi命令显示的磁盘顺序，逐一对磁盘设备生成一条规则，需要在盘符顺序正确时运行，自动生成需要的udev rule文件，并刷新规则：

#!/bin/sh

RULE\_FILE="/etc/udev/rules.d/99-order\_serial\_id.rules"

if [ -f "$RULE\_FILE" ];then

cat /dev/null > $RULE\_FILE

fi

for dev in `lsscsi | grep "/dev/sd" | awk -F " " '{print $NF}'`

do

DEVNUM=`echo ${dev: 7} | tr a-z A-Z`

SERIAL=`udevadm info -q all -n $dev | grep ID\_SERIAL= | awk -F "=" '{print $NF }'`

echo ACTION==\"add\|change\",KERNEL==\"sd\*\",ENV{ID\_SERIAL}==\"$SERIAL\",SYMLINK+=\"sd$DEVNUM%n\" >>$RULE\_FILE

done

/sbin/udevadm control --reload-rules

/sbin/udevadm trigger --type=devices --action=change

以上脚本可以酌情进行修改，如更改为绑定WWID/PATH等。

规则删除脚本：

# vim remove-udev-persistent\_disk\_set.sh

#!/bin/sh

RULE\_FILE="/etc/udev/rules.d/99-order\_serial\_id.rules"

if [ -f "$RULE\_FILE" ];then

   rm -f  $RULE\_FILE

fi

/sbin/udevadm control --reload-rules

/sbin/udevadm trigger --type=devices --action=change

注：实质上使用udev绑定规则建立的新设备名称，和系统下默认已有的/dev/disk/by-id、by-path等软连接名称是同样的原理，也可以无需建立规则，直接使用该名称永久表示磁盘设备。

# ls –ltr /dev/disk/by-id

lrwxrwxrwx. 1 root root 9 Nov 3 13:41 scsi-3600605b00a086f80236fd90e3c1ab21a -> ../../sda

lrwxrwxrwx. 1 root root 10 Nov 3 13:41 scsi-3600605b00a086f80236fd90e3c1ab21a-part1 -> ../../sda1

lrwxrwxrwx. 1 root root 10 Nov 3 13:41 scsi-3600605b00a086f80236fd90e3c1ab21a-part2 -> ../../sda2

lrwxrwxrwx. 1 root root 9 Nov 3 13:41 scsi-3600605b00a086f80236fd91f3d22d2fe -> ../../sdb

lrwxrwxrwx. 1 root root 9 Nov 3 13:41 scsi-3600605b00a086f80236fd92d3dfdcb85 -> ../../sdc

……

# ls –ltr /dev/disk/by-path

lrwxrwxrwx. 1 root root 9 Nov 3 13:41 pci-0000:86:00.0-scsi-0:2:0:0 -> ../../sda

lrwxrwxrwx. 1 root root 10 Nov 3 13:41 pci-0000:86:00.0-scsi-0:2:0:0-part1 -> ../../sda1

lrwxrwxrwx. 1 root root 10 Nov 3 13:41 pci-0000:86:00.0-scsi-0:2:0:0-part2 -> ../../sda2

lrwxrwxrwx. 1 root root 9 Nov 3 13:41 pci-0000:86:00.0-scsi-0:2:1:0 -> ../../sdb

lrwxrwxrwx. 1 root root 9 Nov 3 13:41 pci-0000:86:00.0-scsi-0:2:2:0 -> ../../sdc

……

## fdisk -l 和 lsblk 显示磁盘大小不同

【**问题描述**】

一块10T的希捷硬盘，使用fdisk -l命令查看，显示10000.8GB，而使用lsblk命令显示的是9.1T。

【**原因说明**】

**正常现象，**lsblk和multipath计算磁盘大小使用ioctl,单位为GiB（除以1024）,fdisk计算磁盘大小的单位是GB(除以1000）。

## 如何在安装过程中清理磁盘



在如上安装过程中的安装信息摘要图形界面，使用ctrl+alt+f2组合键切换终端进入命令行界面：



在命令行界面使用ls /dev/sd\* 命令查看磁盘；

使用wipefs -af /dev/sda 命令清理磁盘（其中/dev/sda需要改成对应的盘符）。

清理完成后使用ctrl+alt+f6切换回安装界面继续安装系统。

# **网络部分**

## avahi-daemon: Host name conflict, retrying with linux-2

【**问题描述**】

系统日志中存在下面的报错：

avahi-daemon [11545]: Host name conflict, retrying with linux-2

【**原因说明**】

启动avahi-daemon服务后，检测到在局域网或集群网络中，有和当前主机名冲突的名称，需要变更host name; 若不在局域网或集群环境中，关闭avahi-daemon服务即可。

【**解决方案**】

1. 使用hostnamectl set-hostname “new hostname” 修改主机名
2. 若不在局域网或集群环境中，关闭avahi-daemon服务即可

systemctl stop avahi-daemon

systemctl disable avahi-daemon

## Activation via systemd failed for unit 'dbus-org.freedesktop.resolve1.service': Unit dbus-org.freedesktop.resolve1.service not found.

【问题描述】

系统日志中存在下面的报错：

localhost dbus-daemon[1757]: [system] Activation via systemd failed for unit 'dbus-org.freedesktop.resolve1.service': Unit dbus-org.freedesktop.resolve1.service not found.

【原因说明】

在KOS引导过程中，没有启用systemd-resolved.service服务时，NetworkManager.Service尝试开启dbus-org.freedesktop.resolve1.service服务会出现这个报错,因dbus-org.freedesktop.resolve1.service是systemd-resolved.service的软链接。

【解决方案】

两种解决方案：

方案A：

启用systemd-resolved.service.

# systemctl enable systemd-resolved.service

方案B：不启用systemd-resolved服务

编辑 /etc/NetworkManager/NetworkManager.conf 增加下面的参数

# vi /etc/NetworkManager/NetworkManager.conf

[main]

systemd-resolved=false

# 内存部分

## Out of Memory: Killed process [PID] [process name]

【问题描述】

系统日志中存在下面的报错：

Out of Memory: Killed process [PID] [process name]

【原因说明】

系统触发out of memory告警，系统内存不足，开始根据进程优先级杀低优先级的进程，建议减少消耗内存的应用。

【解决方案】

1.检查进程是否收到cgroup限制；2.检查num内存策略；3.减少消耗内存的应用

## 重启对比/proc/meminfo的内存容量发现内存大小有细微的变化

【问题描述】

系统下多次重启对比/proc/meminfo的内存容量发现，内存大小有细微的变化，

如791740656 kB，可能变化为791740652 kB、791740660 kB

【原因说明】

系统下多次重启看到很少的4K，8k的内存量的变化，可以忽略，是正常的。

【解决方案】

忽略

# 显示部分

## journal: failed to get edid data: EDID length is too small

【问题描述】

系统日志中存在下面的报错：

journal: failed to get edid data: EDID length is too small

【原因说明】

连接的显示器返回的EDID数据块大小（EDID数据块的长度应至少128 bytes）导致的，建议更换显示器。

## failed to get edid: unable to get EDID for output

【问题描述】

系统日志中存在下面的报错：

failed to get edid: unable to get EDID for output

【原因说明】

EDID(扩展显示识别数据)是通过参考伪文件/sys/class/drm/cardX-\*\*\*/edid 得到的，如下所示:

$ cat /sys/class/drm/cardX-\*\*\*/edid

如果此文件的引用出现问题，则会发生该错误。

虚拟机(如KVM)可能无法获得EDID，因为虚拟机不支持显示信息。

【解决方案】

可尝试更换显示设备，若是虚拟机，则无需关注。

# 其他部分

## BMC控制台大小写键盘混乱

【问题描述】

在某些服务器上通过键盘向bmc控制台输入内容时，出现大小写混乱问题。

【原因说明】

KVM里的正确输入是要在打开KVM的窗口中输入，里边有组合键也有小键盘。

不能使用打开KVM的PC机键盘输入热键、组合键等，原因：PC的os会拦截键盘输入的热键、组合键等信息，导致实际KVM收到的输入信息失真，于是表现为乱码。

## 每次重启messages中的启动时间会快8小时，直到Journal启动后恢复正常时间

【问题描述】

每次重启messages中的启动时间会快8小时，直到Journal重启后恢复正常时，比如：

正常时间应是 Dec 20 15:38:14，但当系统重启时，时间会增加8小时，变化如下，initramfs加载完成，到启动kernel重启后，时间由 Dec 20 23:38:12 恢复为Dec 20 15:38:12。

Dec 20 23:38:10 localhost journal: Runtime journal is using 8.0M (max allowed 4.0G, trying to leave 4.0G free of 251.8G available → current limit 4.0G).

Dec 20 23:38:10 localhost kernel: Initializing cgroup subsys cpuset

Dec 20 23:38:10 localhost kernel: Initializing cgroup subsys cpu

Dec 20 23:38:10 localhost kernel: Initializing cgroup subsys cpuacct

Dec 20 23:38:10 localhost kernel: Linux version 4.18.0-372.26.1.kos5.x86\_64 (mockbuild@fa33ebc3beb6429381631eb12e3992e8) (gcc version 8.5.0 20210514 (kos 8.5.0-10.0.1) (GCC) ) #1 SMP Thu Oct 29 17:29:29 EDT 2022

........................

Dec 20 23:38:12 localhost systemd: Starting Cleanup udevd DB...

Dec 20 23:38:12 localhost systemd: Started Cleanup udevd DB.

Dec 20 23:38:12 localhost systemd: Reached target Switch Root.

Dec 20 23:38:12 localhost systemd: Starting Switch Root.

Dec 20 23:38:12 localhost systemd: Starting Switch Root...

Dec 20 23:38:12 localhost systemd: Switching root.

**Dec 20 23:38:12** localhost journal: Journal stopped

**Dec 20 15:38:12** localhost journal: Runtime journal is using 8.0M (max allowed 4.0G, trying to leave 4.0G free of 251.8G available → current limit 4.0G).

Dec 20 15:38:12 localhost journal: Runtime journal is using 8.0M (max allowed 4.0G, trying to leave 4.0G free of 251.8G available → current limit 4.0G).

Dec 20 15:38:12 localhost systemd-journald[321]: Received SIGTERM from PID 1 (systemd).

Dec 20 15:38:12 localhost kernel: SELinux:  Disabled at runtime.

Dec 20 15:38:12 localhost kernel: type=1404 audit(1482248292.393:2): selinux=0 auid=4294967295 ses=4294967295

Dec 20 15:38:12 localhost systemd[1]: **RTC configured in localtime, applying delta of 480 minutes to system time.**

Dec 20 15:38:12 localhost kernel: ip\_tables: (C) 2000-2006 Netfilter Core Team

Dec 20 15:38:12 localhost systemd[1]: Inserted module 'ip\_tables'

Dec 20 15:38:12 localhost journal: Journal started

【原因说明】

由于RTC时间(BIOS时间)默认认为和UTC时间一致，系统重启日志前半段initramfs加载时，由于时区是+8区，会增加480分钟，即8小时；加载到kernel时，系统检测到RTC configured in localtime的设置，Local time应和RTC time一致，将时间调整-8小时；系统运行中，系统时间和RTC时间一致，若再次重启，initramfs启动日志中，仍然会认为BIOS时间是UTC时间，会按时区+8，加载到kernel时，和RTC保持一致，再-8小时，如此反复循环。

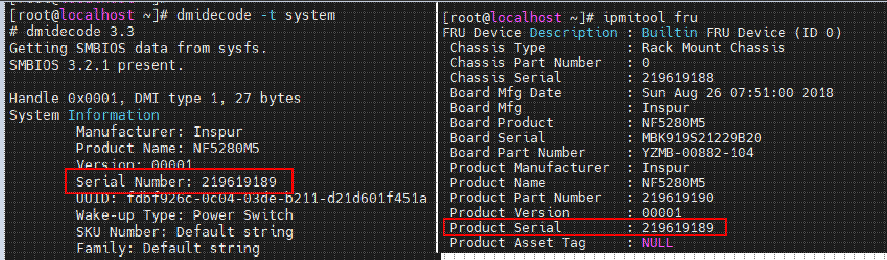
【解决方案】

可以使用如下命令将硬件时钟设置为UTC时间

# timedatectl set-local-rtc 0

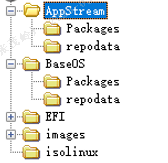
## 如何在KOS系统中读取服务器SN信息

【解决方案】

KOS系统下可以通过dmidecode -t system或者ipmitool fru 命令来查看SN信息，参考如下：

## 如何使用系统镜像配置本地yum源

KOS 5.8系统镜像中将rpm分成两类：一类是基础rpm安装包，在BaseOS文件夹中；一类是与应用相关的rpm，在AppStream文件夹中：



故在做yum源时，需要做两个yum源，一个BaseOS，一个AppStream。 比如将iso镜像文件挂载到了/mnt目录，创建/etc/yum.repos.d/local.repo配置文件，内容如下：

