

Linux服务器构建与运维管理

从基础到实战（基于 openEuler）

第11章：运维管理

阮晓龙

13938213680 / ruanxiaolong@hactcm.edu.cn

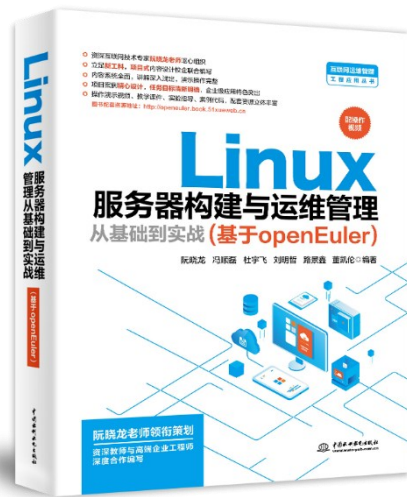
<https://internet.hactcm.edu.cn>
<http://www.51xueweb.cn>

河南中医药大学信息技术学院互联网技术教学团队
河南中医药大学医疗健康信息工程技术研究所

2024.11

提纲

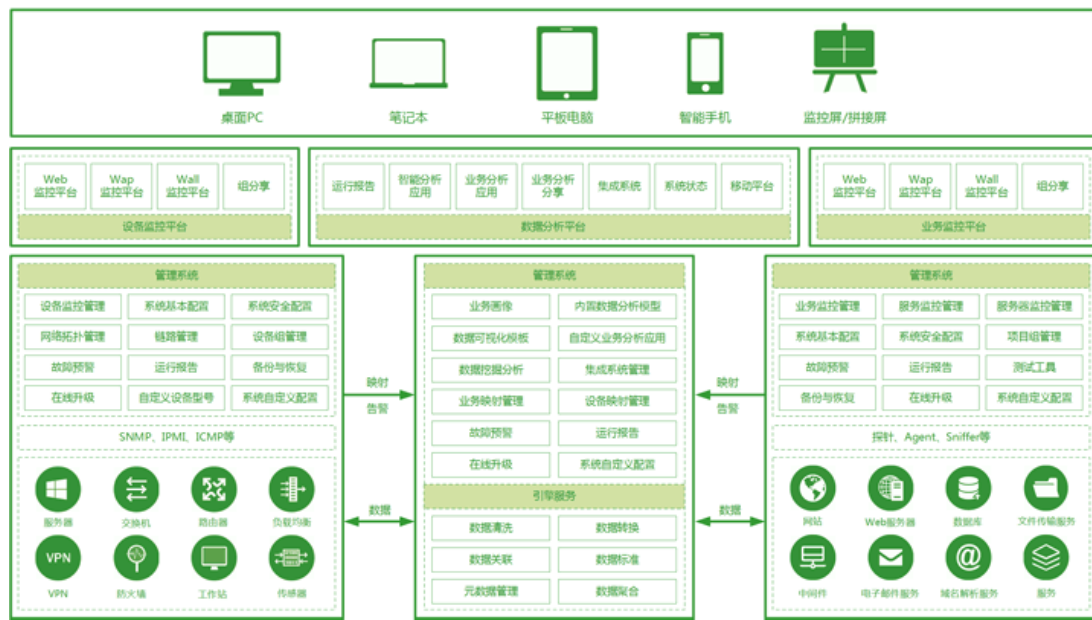
- 了解系统运维管理
- 查看本地主机的信息
 - 查看系统硬件信息
 - 查看系统存储使用情况
 - 查看网络通信情况
- 监控本地主机的性能
 - 使用sysstat工具监控性能
 - 访问PROC获取指标数据
- 实现可视化系统监控
 - Linux-Dash
 - Monitorix



1. 了解系统运维管理

1.1 什么是运维

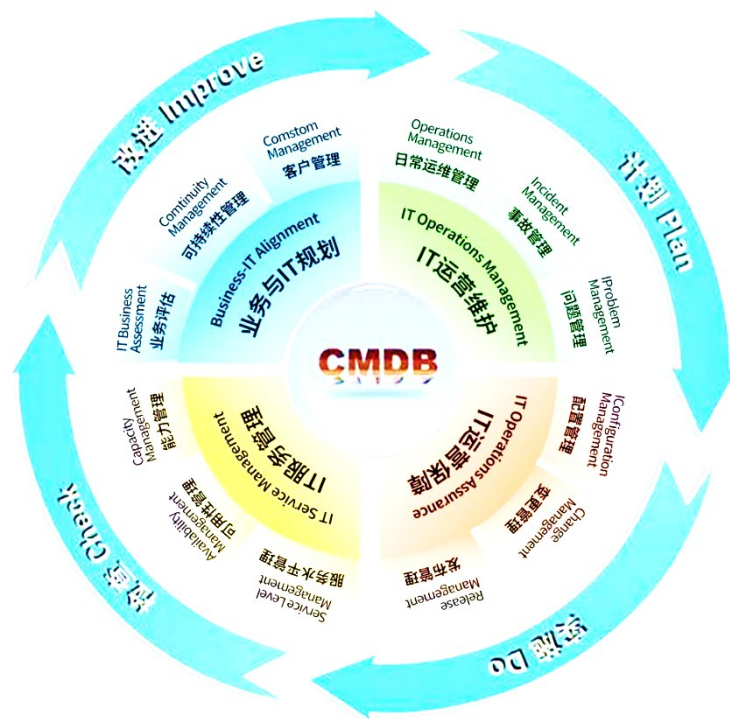
- 运维管理是对系统运行状态进行控制，快速响应并调整业务运行性能等，使之与业务运行的预期目标一致，实现对操作系统未来发展趋势的维护和保障。



1. 了解系统运维管理

1.1 什么是运维

- 深入了解系统的运行状况，持续保障业务的稳定运行是运维管理的基本内容。
- 运维管理不是一个技术措施，而是长期持续的系统工作。



1. 了解系统运维管理

1.2 操作系统运维的内容

- 操作系统运维管理的基本内容：
 - 系统运行监控。
 - 负责查看服务器硬件信息及运行状态，了解操作系统的基本信息以及所安装部署的业务情况等。
 - 权限管理。
 - 负责了解系统用户的权限设置、为新用户增设账号、将不再活动的用户删除、将近期不再访问的用户禁用等账号处理相关事务，及时更新系统权限配置，保障系统用户权限安全。
 - CPU 管理。
 - 负责监控 CPU 的负载情况，优化资源利用，降低系统负载压力。
 - 内存管理。
 - 负责监控业务系统内存、缓存、交换空间等方面的使用情况，合理调配业务资源，保障业务高性能运转。



1. 了解系统运维管理

1.2 操作系统运维的内容

- 操作系统运维管理的基本内容：
 - 磁盘管理。
 - 负责检查硬件磁盘的运行状态、及时更换物理磁盘并配置系统能够识别新的磁盘信息，从而使系统使用新增的存储资源。负责查看磁盘的使用情况、了解磁盘的 IO 读写速率、利用率、吞吐量等指标运行情况，保障业务数据的存储效率。
 - 网络管理。
 - 负责了解主机的网络流量，合理规划网络结构，能够及时发现网络故障并做出响应与解决。
 - 进程管理。
 - 负责查看系统的相关进程信息，处理系统无用进程占用系统资源，降低系统运行负载。
 - 日志管理。
 - 合理记录系统日志，便于操作追溯和日志审查分析。

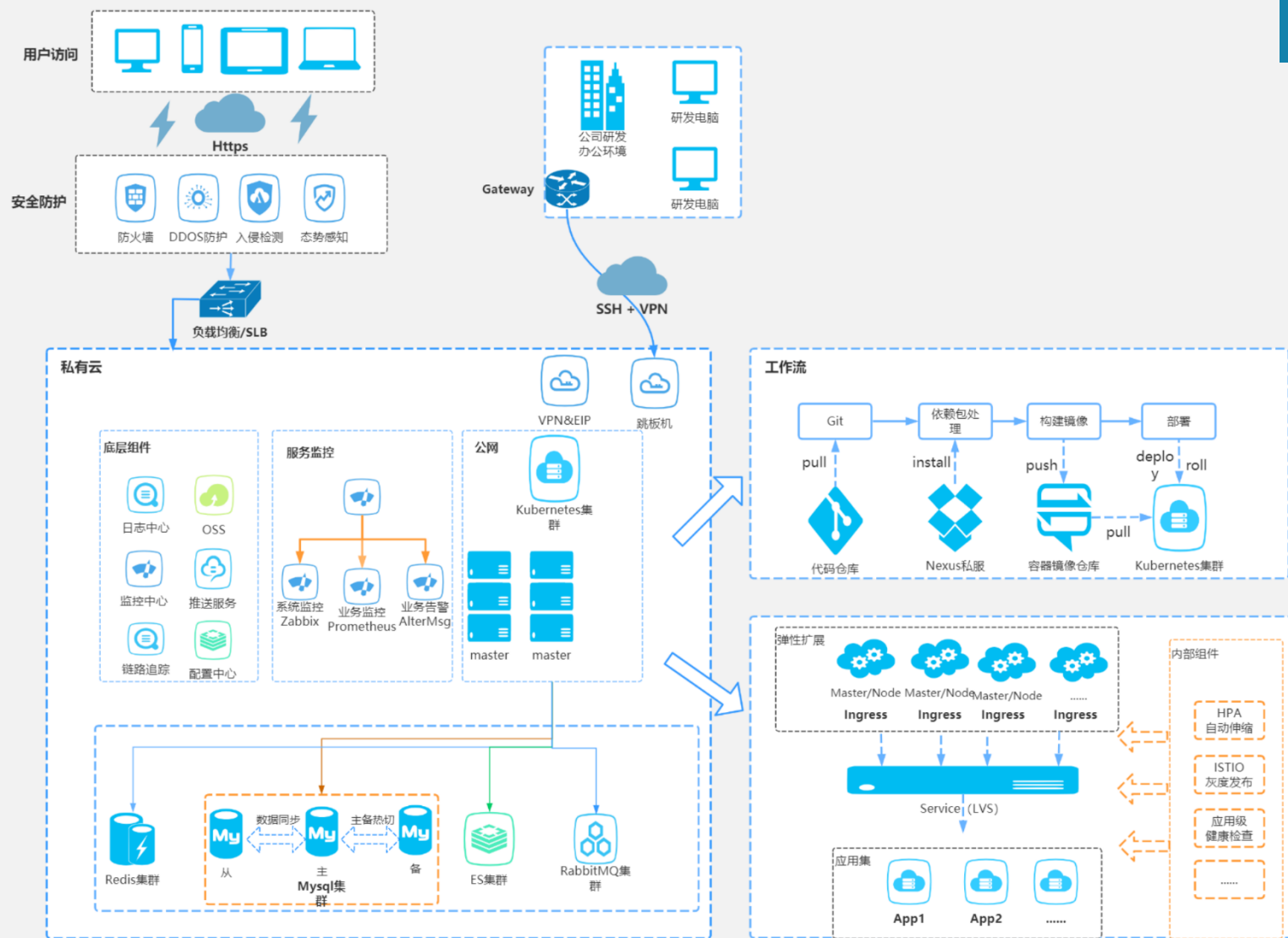


1. 了解系统运维管理

1.3 操作系统运维的内容

- 系统运维管理的方式可分为命令管理和自动化管理。
 - 命令管理。
 - 通过操作系统的命令实现系统配置管理。
 - 常用的管理命令有 vi（对文件进行编辑管理）、fdisk（对磁盘进行管理）、nmcli（对网络进行管理）、systemctl（对服务进行管理）等。
 - 自动化管理。
 - 通过自动化运维工具实现对批量主机进行配置管理，实现对系统的网络、存储、应用交付等自动化配置，降低运维管理人员的压力，减少或避免重复性工作。







建_的成 / 看得见 / 管_的住 / 用_的好



1. 了解系统运维管理

- 系统监控的产生缘由：
 - 随着信息化建设不断深入，应用系统不断增多，运维人员管理的设备、业务数量也急剧增加，如何直观地查看多个设备、业务的运行情况，并保证出现异常时能及时发现，成为运维人员最关心也需要迫切解决的问题。
- 通过系统监控可以：
 - 实时了解系统状态
 - 快速发现系统异常
 - 及时解决异常问题
 - 保障可靠稳定服务
- 系统监控的内容：
 - 系统监控是对操作系统整体运行情况的监控。
 - 通常监控系统的 CPU、物理内存、虚拟内存、进程、存储、网络等运行状态。



1. 了解系统运维管理

1.4 操作系统的监控

- 按照监控实现方式不同，系统监控可分为命令监控和软件监控两类。
 - 命令监控
 - 通过操作系统的命令实现对系统运行情况的监控。
 - 常用的监控命令有 top（查看所有正在运行且处于活动状态的实时进程）、netstat（查看系统网络性能情况）、iostat查看系统CPU使用情况与磁盘I/O情况）等。
 - 软件监控
 - 通过专用的监控软件，借助简单网络管理协议（Simple Network Management Protocol, SNMP）、Agent、探针等手段，对系统运行情况进行周期性监控，记录监控数据，实现监控历史数据查看及系统运行情况分析，并将系统异常情况通过某种方式（如电子邮件、短信、微信、App 等）通知相关人员。



监控设备数
39监控业务数
102故障数量
0预警数量
0日志总量
2.65亿日志告警
0

设备状态



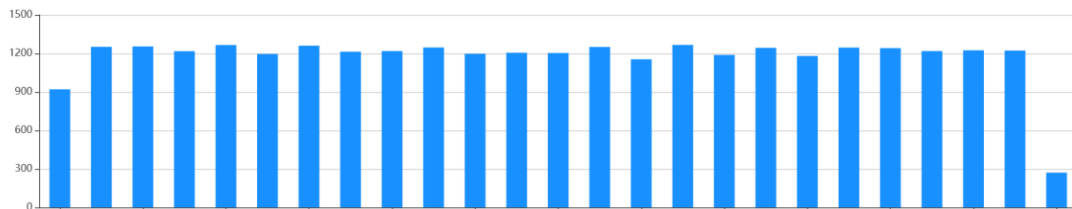
业务状态



业务故障预警

序号	名称	故障预警原因	持续时间	操作
1	QNH-URL-NSM	Connection timed out af...	1分	解决方案 提交报修
2	QNH-URL-NSM	Connection timed out af...	6分	解决方案 提交报修
3	QNH-URL-NSM	Connection timed out af...	3分	解决方案 提交报修
4	QNH-URL-NSM	Connection timed out af...	3分	解决方案 提交报修
5	QNH-URL-NSM	Connection timed out af...	3分	解决方案 提交报修
6	QNH-URL-NSM	Connection timed out af...	3分	解决方案 提交报修
7	QNH-URL-NSM	Connection timed out af...	3分	解决方案 提交报修
8	QNH-URL-NSM	Connection timed out af...	3分	解决方案 提交报修

最近24小时日志



场景分析



数据中心分析

总览数据中心网络、设备、业务运行情况

[查看详情](#)

数据中心透视

总览当前告警、网络设备、服务器运行情况

[查看详情](#)

网络运行分析

总览关键网络接口通信情况与流量状态

[查看详情](#)

机柜落位分析

总览机柜落位与使用情况、资产状态及告警统计

[查看详情](#)

运维工单分析

总览工单状态, 包括工单统计、处理人工量

[查看详情](#)

日志数据分析

总览日志类型、数量、来源统计与告警消息

[查看详情](#)

集成系统



运维管理综合服务体系

提供资产管理、事件报修服务, 基于ITIL处理流程管理运维工作

[查看详情](#)

设备运维监控分析系统

实现全设备运行状态监控, 提供性能分析、告警通知与运行报告

[查看详情](#)

业务运维监控分析系统

实现全业务运行状态监控, 提供性能分析、告警通知与运行报告

[查看详情](#)

数据中心综合管理系统

基于IT资产管理与监控, 提供数据中心一体化运维监控管理服务

[查看详情](#)

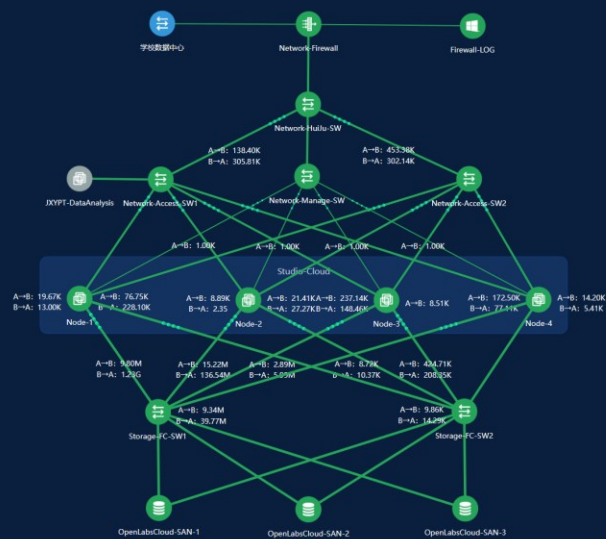
日志数据智能分析系统

[查看详情](#)

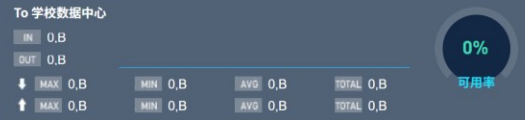
健康度



拓扑结构



出口带宽使用情况



服务器运行状态



Web 服务运行状态



2. 查看本地主机的信息

操作系统的信息要看什么？

计算

Hardware
CPU

存储

Memory
Disk

网络

Traffic
connectivity



2. 查看本地主机的信息

2.1 查看系统硬件信息



dmidecode [选项]

功能:

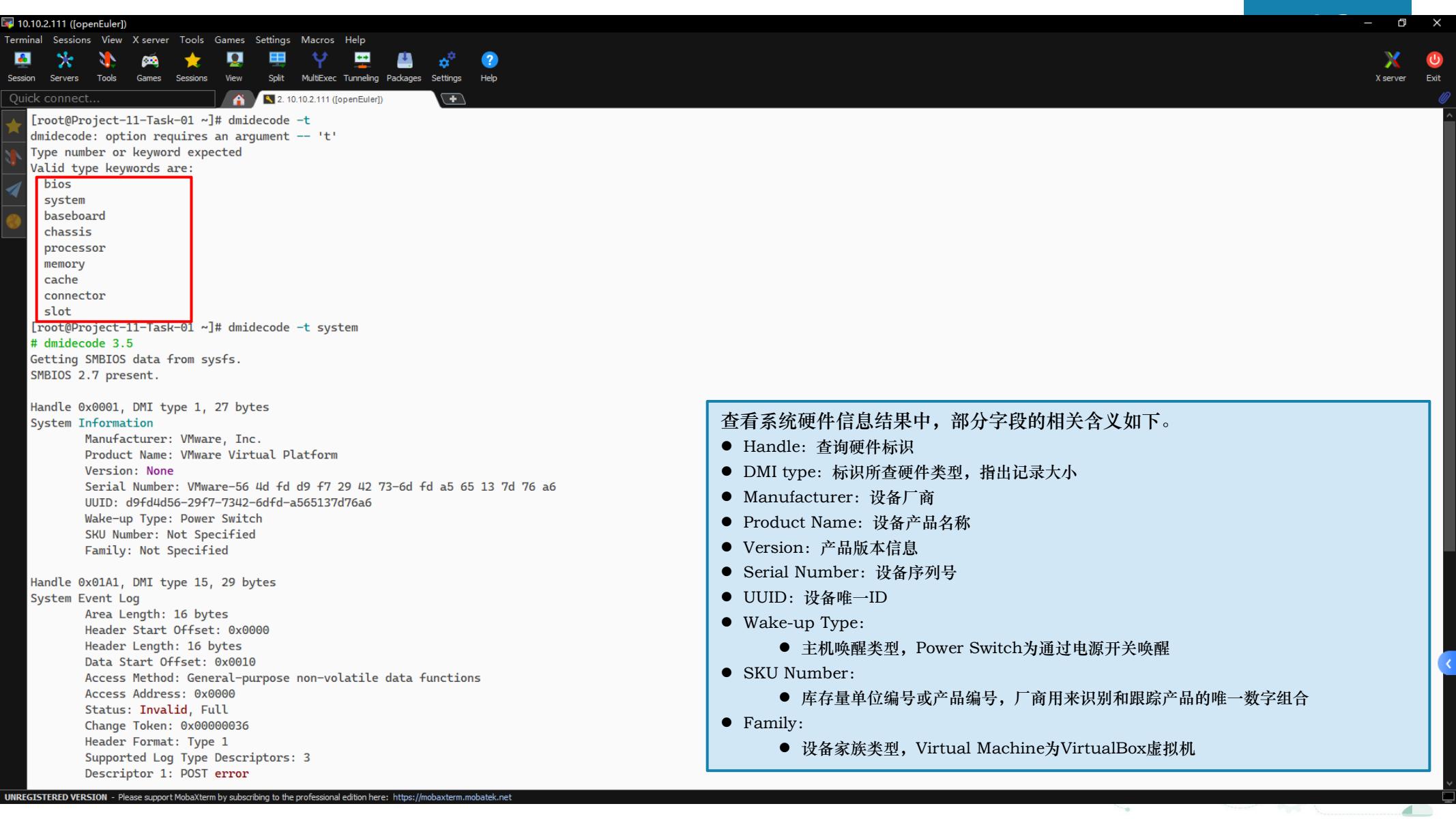
- 查看系统硬件信息。

参数/命令:

主要选项:

- 空: 输出所有硬件信息
- -q: 输出硬件信息, 比较简洁
- -d: 从设备文件读取信息
- -s: 只显示指定 DMI 字符串的信息
- -t bios: 查看 BIOS 相关的硬件信息
- -t system: 查看系统相关的硬件信息
- -t baseboard: 查看主板相关的硬件信息
- -t chassis: 查看机箱相关的硬件信息
- -t processor: 查看处理器相关的硬件信息
- -t memory: 查看内存相关的硬件信息
- -t cache: 查看缓存的相关信息
- -t connector: 查看端口连接器的相关信息
- -t slot: 查看系统槽的相关信息





```
[root@Project-11-Task-01 ~]# dmidecode -t
dmidecode: option requires an argument -- 't'
Type number or keyword expected
Valid type keywords are:
```

- bios
- system
- baseboard
- chassis
- processor
- memory
- cache
- connector
- slot

```
[root@Project-11-Task-01 ~]# dmidecode -t system
# dmidecode 3.5
Getting SMBIOS data from sysfs.
SMBIOS 2.7 present.
```

```
Handle 0x0001, DMI type 1, 27 bytes
System Information
Manufacturer: VMware, Inc.
Product Name: VMware Virtual Platform
Version: None
Serial Number: VMware-56 4d fd d9 f7 29 42 73-6d fd a5 65 13 7d 76 a6
UUID: d9fd4d56-29f7-7342-6dfd-a565137d76a6
Wake-up Type: Power Switch
SKU Number: Not Specified
Family: Not Specified
```

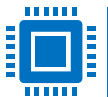
```
Handle 0x01A1, DMI type 15, 29 bytes
System Event Log
Area Length: 16 bytes
Header Start Offset: 0x0000
Header Length: 16 bytes
Data Start Offset: 0x0010
Access Method: General-purpose non-volatile data functions
Access Address: 0x0000
Status: Invalid, Full
Change Token: 0x00000036
Header Format: Type 1
Supported Log Type Descriptors: 3
Descriptor 1: POST error
```

查看系统硬件信息结果中，部分字段的相关含义如下。

- Handle: 查询硬件标识
- DMI type: 标识所查硬件类型，指出记录大小
- Manufacturer: 设备厂商
- Product Name: 设备产品名称
- Version: 产品版本信息
- Serial Number: 设备序列号
- UUID: 设备唯一ID
- Wake-up Type:
 - 主机唤醒类型，Power Switch为通过电源开关唤醒
- SKU Number:
 - 库存量单位编号或产品编号，厂商用来识别和跟踪产品的唯一数字组合
- Family:
 - 设备家族类型，Virtual Machine为VirtualBox虚拟机

2. 查看本地主机的信息

2.1 查看系统硬件信息



lspci [选项]

功能:

- 总览主机硬件设备列表信息。

参数/命令:

主要选项:

- 空: 输出所有 PCI 硬件设备信息
- -t:
 - 以树形方式显示所有总线、桥、设备和PCI连接的关系
- -b: 以总线为中心进行查看
- -d [<vendor>]:
 - 只显示指定生产厂商和设备 ID 的设备
- -n: 以数字形式显示 PCI 生产厂商和设备号
- -s [[<bus>]:[<slot>][. [<func>]]:
 - 显示指定总线、插槽上的设备或设备上的功能块信息
- -i <file>:
 - 指定 PCI 编号列表文件
- -m: 以可读的方式显示 PCI 设备信息
- -v: 以冗余模式显示所有设备的详细信息
- -vv: 以更冗余模式显示所有设备的更详细的信息

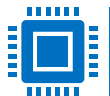
```
[root@Project-11-Task-01 ~]# lspci
00:00.0 Host bridge: Intel Corporation 440BX/ZX/DX - 82443BX/ZX/DX Host bridge (rev 01)
00:01.0 PCI bridge: Intel Corporation 440BX/ZX/DX - 82443BX/ZX/DX AGP bridge (rev 01)
00:07.0 ISA bridge: Intel Corporation 82371AB/EB/MB PIIX4 ISA (rev 08)
00:07.1 IDE interface: Intel Corporation 82371AB/EB/MB PIIX4 IDE (rev 01)
00:07.3 Bridge: Intel Corporation 82371AB/EB/MB PIIX4 ACPI (rev 08)
00:07.7 System peripheral: VMware Virtual Machine Communication Interface (rev 10)
00:0f.0 VGA compatible controller: VMware SVGA II Adapter
00:10.0 SCSI storage controller: Broadcom / LSI 53c1030 PCI-X Fusion-MPT Dual Ultra320 SCSI (rev 01)
00:11.0 PCI bridge: VMware PCI bridge (rev 02)
00:15.0 PCI bridge: VMware PCI Express Root Port (rev 01)
00:15.1 PCI bridge: VMware PCI Express Root Port (rev 01)
00:15.2 PCI bridge: VMware PCI Express Root Port (rev 01)
00:15.3 PCI bridge: VMware PCI Express Root Port (rev 01)
00:15.4 PCI bridge: VMware PCI Express Root Port (rev 01)
00:15.5 PCI bridge: VMware PCI Express Root Port (rev 01)
00:15.6 PCI bridge: VMware PCI Express Root Port (rev 01)
00:15.7 PCI bridge: VMware PCI Express Root Port (rev 01)
00:16.0 PCI bridge: VMware PCI Express Root Port (rev 01)
00:16.1 PCI bridge: VMware PCI Express Root Port (rev 01)
00:16.2 PCI bridge: VMware PCI Express Root Port (rev 01)
00:16.3 PCI bridge: VMware PCI Express Root Port (rev 01)
00:16.4 PCI bridge: VMware PCI Express Root Port (rev 01)
00:16.5 PCI bridge: VMware PCI Express Root Port (rev 01)
00:16.6 PCI bridge: VMware PCI Express Root Port (rev 01)
00:16.7 PCI bridge: VMware PCI Express Root Port (rev 01)
00:17.0 PCI bridge: VMware PCI Express Root Port (rev 01)
00:17.1 PCI bridge: VMware PCI Express Root Port (rev 01)
00:17.2 PCI bridge: VMware PCI Express Root Port (rev 01)
00:17.3 PCI bridge: VMware PCI Express Root Port (rev 01)
00:17.4 PCI bridge: VMware PCI Express Root Port (rev 01)
00:17.5 PCI bridge: VMware PCI Express Root Port (rev 01)
00:17.6 PCI bridge: VMware PCI Express Root Port (rev 01)
00:17.7 PCI bridge: VMware PCI Express Root Port (rev 01)
00:18.0 PCI bridge: VMware PCI Express Root Port (rev 01)
00:18.1 PCI bridge: VMware PCI Express Root Port (rev 01)
00:18.2 PCI bridge: VMware PCI Express Root Port (rev 01)
00:18.3 PCI bridge: VMware PCI Express Root Port (rev 01)
00:18.4 PCI bridge: VMware PCI Express Root Port (rev 01)
00:18.5 PCI bridge: VMware PCI Express Root Port (rev 01)
00:18.6 PCI bridge: VMware PCI Express Root Port (rev 01)
00:18.7 PCI bridge: VMware PCI Express Root Port (rev 01)
```

查看主板硬件信息结果中，部分字段的相关含义如下。

- 00:00.0: 位于PCI总线上的位置，其中00表示总线号，00表示设备号，0表示功能号
- Host bridge: 主机桥接器，用于连接CPU和其他PCI设备的桥接器
- Intel Corporation 440BX/ZX/DX: 制造商是Intel Corporation，型号为440
- 82443BX/ZX/DX: 表示该设备所属的子系统型号为82443BX/ZX/DX
- rev 01: 表示该设备的版本号为02

2. 查看本地主机的信息

2.1 查看系统硬件信息



lshw [选项]

功能:

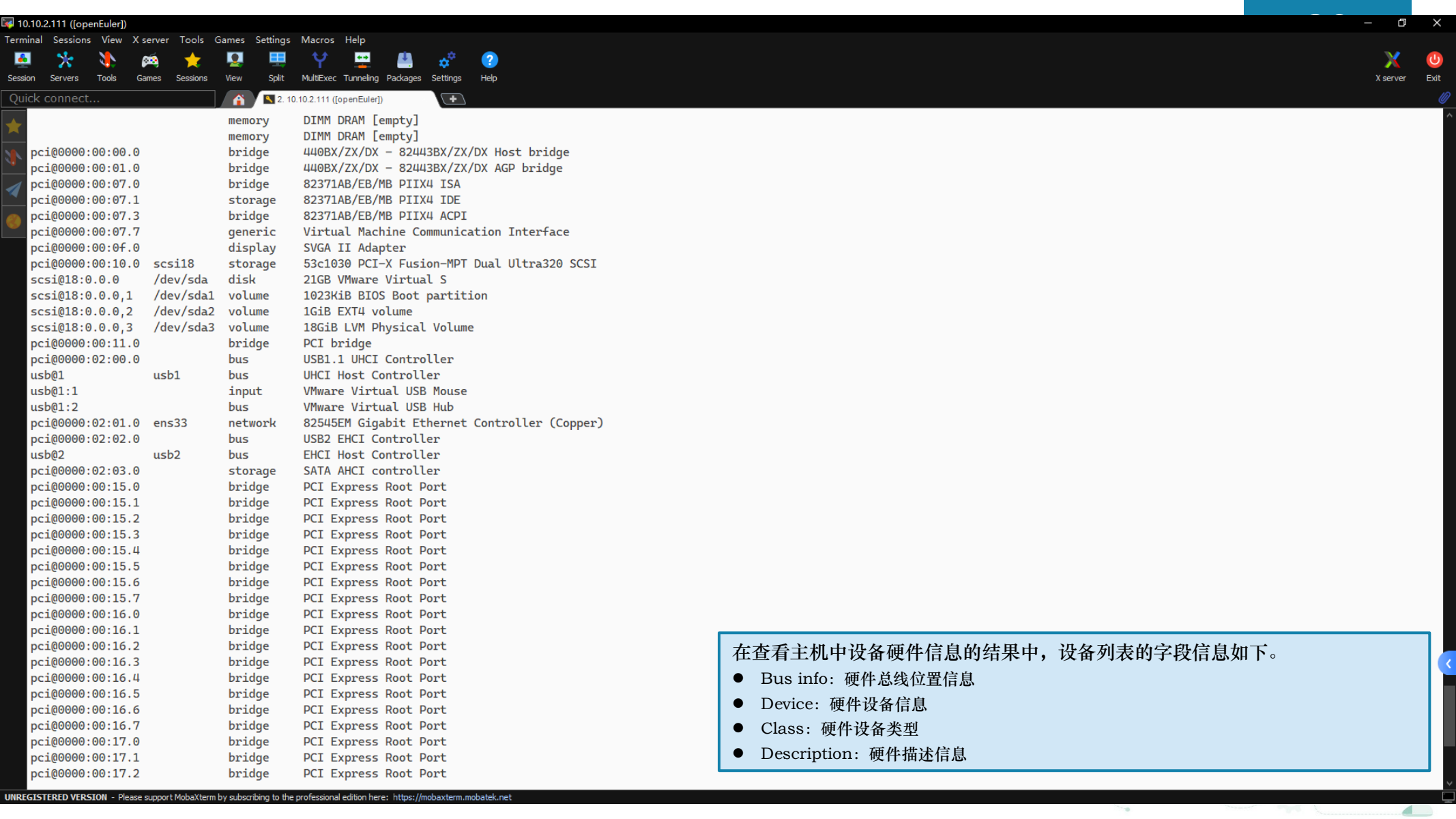
- 总览主机硬件设备的列表信息。

参数/命令:

主要选项:

- 空: 输出所有硬件设备信息
- -short:
 - 显示设备列表, 输出包括设备路径(path)、类别(class)以及简单描述
- -businfo:
 - 显示设备列表, 输出包括总线信息、SCSI、USB、IDE、PCI 地址等
- -C <Class>:
 - 根据类型查看相应的设备信息

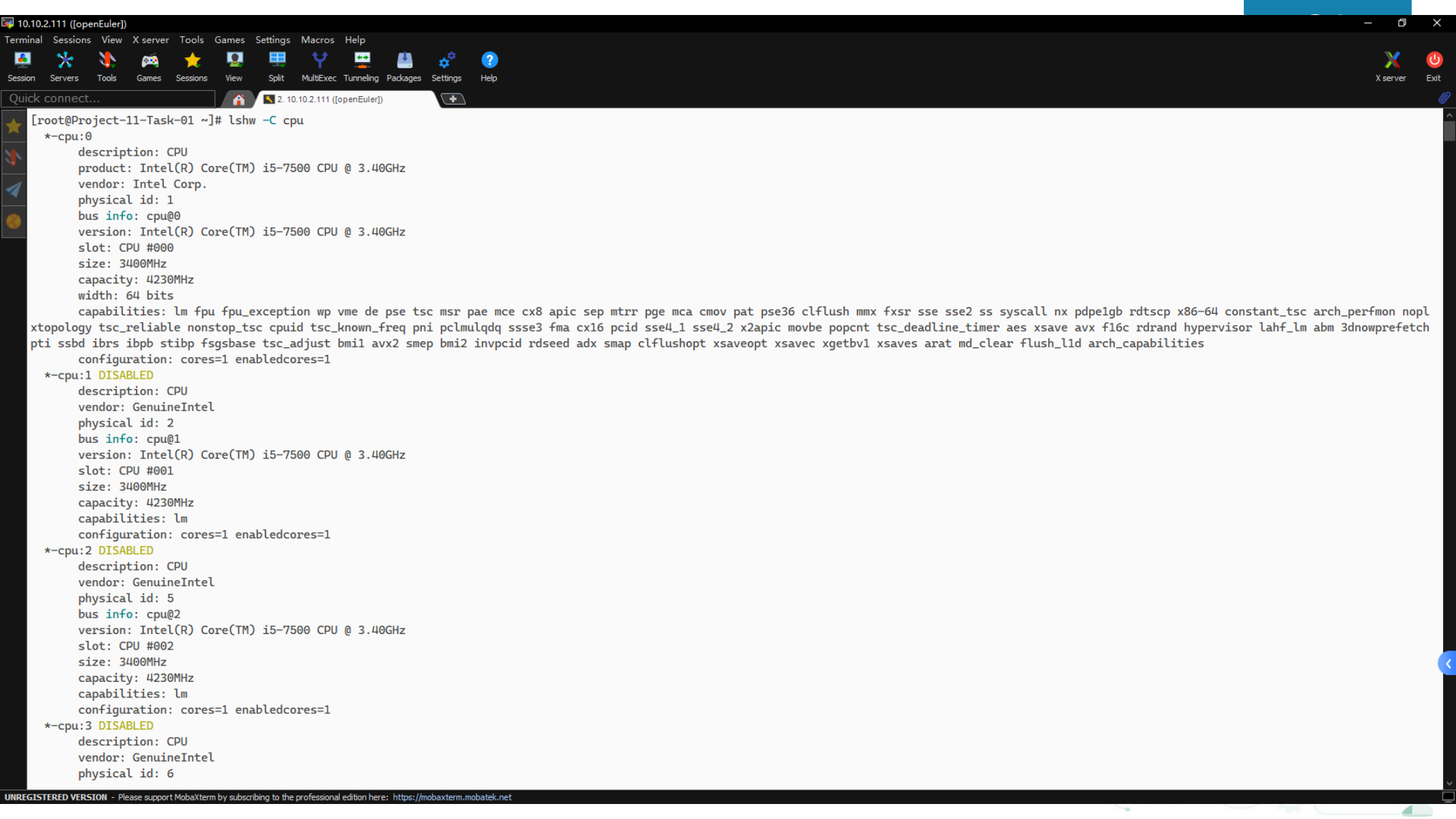




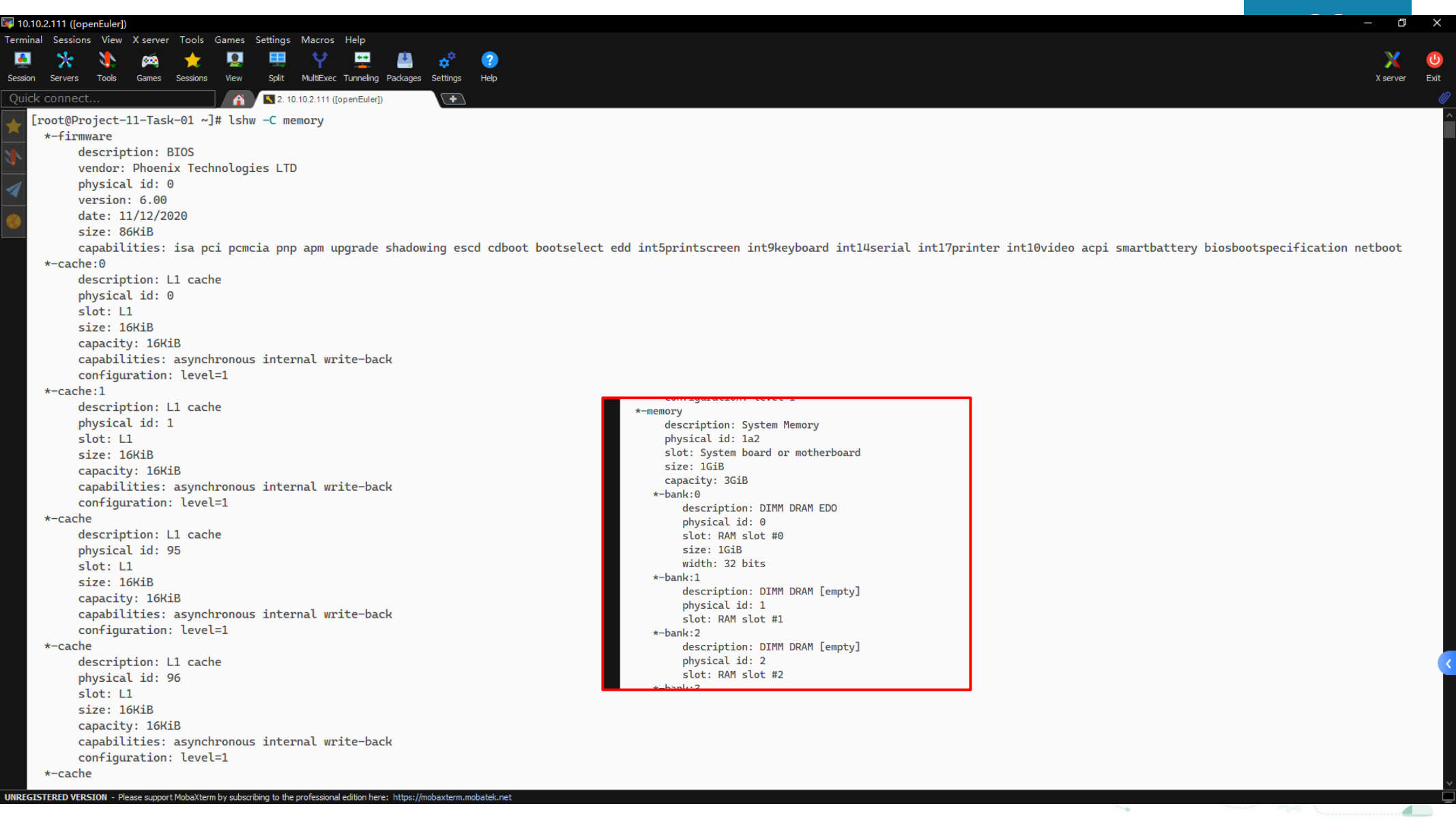
```
memory DIMM DRAM [empty]
memory DIMM DRAM [empty]
pci@0000:00:00.0 bridge 440BX/ZX/DX - 82443BX/ZX/DX Host bridge
pci@0000:00:01.0 bridge 440BX/ZX/DX - 82443BX/ZX/DX AGP bridge
pci@0000:00:07.0 bridge 82371AB/EB/MB PIIX4 ISA
pci@0000:00:07.1 storage 82371AB/EB/MB PIIX4 IDE
pci@0000:00:07.3 bridge 82371AB/EB/MB PIIX4 ACPI
pci@0000:00:07.7 generic Virtual Machine Communication Interface
pci@0000:00:0f.0 display SVGA II Adapter
pci@0000:00:10.0 scsi18 storage 53c1030 PCI-X Fusion-MPT Dual Ultra320 SCSI
scsi@18:0.0.0 /dev/sda disk 21GB VMware Virtual S
scsi@18:0.0.0,1 /dev/sda1 volume 1023KiB BIOS Boot partition
scsi@18:0.0.0,2 /dev/sda2 volume 1GiB EXT4 volume
scsi@18:0.0.0,3 /dev/sda3 volume 18GiB LVM Physical Volume
pci@0000:00:11.0 bridge PCI bridge
pci@0000:02:00.0 bus USB1.1 UHCI Controller
usb@1 usb1 bus UHCI Host Controller
usb@1:1 input VMware Virtual USB Mouse
usb@1:2 bus VMware Virtual USB Hub
pci@0000:02:01.0 ens33 network 82545EM Gigabit Ethernet Controller (Copper)
pci@0000:02:02.0 bus USB2 EHCI Controller
usb@2 usb2 bus EHCI Host Controller
pci@0000:02:03.0 storage SATA AHCI controller
pci@0000:00:15.0 bridge PCI Express Root Port
pci@0000:00:15.1 bridge PCI Express Root Port
pci@0000:00:15.2 bridge PCI Express Root Port
pci@0000:00:15.3 bridge PCI Express Root Port
pci@0000:00:15.4 bridge PCI Express Root Port
pci@0000:00:15.5 bridge PCI Express Root Port
pci@0000:00:15.6 bridge PCI Express Root Port
pci@0000:00:15.7 bridge PCI Express Root Port
pci@0000:00:16.0 bridge PCI Express Root Port
pci@0000:00:16.1 bridge PCI Express Root Port
pci@0000:00:16.2 bridge PCI Express Root Port
pci@0000:00:16.3 bridge PCI Express Root Port
pci@0000:00:16.4 bridge PCI Express Root Port
pci@0000:00:16.5 bridge PCI Express Root Port
pci@0000:00:16.6 bridge PCI Express Root Port
pci@0000:00:16.7 bridge PCI Express Root Port
pci@0000:00:17.0 bridge PCI Express Root Port
pci@0000:00:17.1 bridge PCI Express Root Port
pci@0000:00:17.2 bridge PCI Express Root Port
```

在查看主机中设备硬件信息的结果中，设备列表的字段信息如下。

- Bus info: 硬件总线位置信息
- Device: 硬件设备信息
- Class: 硬件设备类型
- Description: 硬件描述信息



```
[root@Project-11-Task-01 ~]# lshw -C cpu
*-cpu:0
  description: CPU
  product: Intel(R) Core(TM) i5-7500 CPU @ 3.40GHz
  vendor: Intel Corp.
  physical id: 1
  bus info: cpu@0
  version: Intel(R) Core(TM) i5-7500 CPU @ 3.40GHz
  slot: CPU #000
  size: 3400MHz
  capacity: 4230MHz
  width: 64 bits
  capabilities: lm fpu fpu_exception wp vme de pse tsc msr pae mce cx8 apic sep mtrr pge mca cmov pat pse36 clflush mmx fxsr sse sse2 ss syscall nx pdpe1gb rdtscp x86-64 constant_tsc arch_perfmon nopl
xtopology tsc_reliable nonstop_tsc cpuid tsc_known_freq pni pclmulqdq sse3 fma cx16 pcid sse4_1 sse4_2 x2apic movbe popcnt tsc_deadline_timer aes xsave avx f16c rdrand hypervisor lahf_lm abm 3dnowprefetch
pti ssbd ibrs ibpb stibp fsgsbase tsc_adjust bmi1 avx2 smep bmi2 invpcid rdseed adx smap clflushopt xsaveopt xsavec xgetbv1 xsaves arat md_clear flush_l1d arch_capabilities
  configuration: cores=1 enabledcores=1
*-cpu:1 DISABLED
  description: CPU
  vendor: GenuineIntel
  physical id: 2
  bus info: cpu@1
  version: Intel(R) Core(TM) i5-7500 CPU @ 3.40GHz
  slot: CPU #001
  size: 3400MHz
  capacity: 4230MHz
  capabilities: lm
  configuration: cores=1 enabledcores=1
*-cpu:2 DISABLED
  description: CPU
  vendor: GenuineIntel
  physical id: 5
  bus info: cpu@2
  version: Intel(R) Core(TM) i5-7500 CPU @ 3.40GHz
  slot: CPU #002
  size: 3400MHz
  capacity: 4230MHz
  capabilities: lm
  configuration: cores=1 enabledcores=1
*-cpu:3 DISABLED
  description: CPU
  vendor: GenuineIntel
  physical id: 6
```



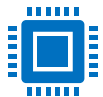
```

[root@Project-11-Task-01 ~]# lshw -C memory
*-firmware
  description: BIOS
  vendor: Phoenix Technologies LTD
  physical id: 0
  version: 6.00
  date: 11/12/2020
  size: 86KiB
  capabilities: isa pci pcmcia pnp apm upgrade shadowing escd cdboot bootselect edd int5pinctscreen int9keyboard int14serial int17printer int10video acpi smartbattery biosbootspecification netboot
*-cache:0
  description: L1 cache
  physical id: 0
  slot: L1
  size: 16KiB
  capacity: 16KiB
  capabilities: asynchronous internal write-back
  configuration: level=1
*-cache:1
  description: L1 cache
  physical id: 1
  slot: L1
  size: 16KiB
  capacity: 16KiB
  capabilities: asynchronous internal write-back
  configuration: level=1
*-cache
  description: L1 cache
  physical id: 95
  slot: L1
  size: 16KiB
  capacity: 16KiB
  capabilities: asynchronous internal write-back
  configuration: level=1
*-cache
  description: L1 cache
  physical id: 96
  slot: L1
  size: 16KiB
  capacity: 16KiB
  capabilities: asynchronous internal write-back
  configuration: level=1
*-memory
  description: System Memory
  physical id: 1a2
  slot: System board or motherboard
  size: 1GiB
  capacity: 3GiB
  *-bank:0
    description: DIMM DRAM EDO
    physical id: 0
    slot: RAM slot #0
    size: 1GiB
    width: 32 bits
  *-bank:1
    description: DIMM DRAM [empty]
    physical id: 1
    slot: RAM slot #1
  *-bank:2
    description: DIMM DRAM [empty]
    physical id: 2
    slot: RAM slot #2
  *-bank:3

```

2. 查看本地主机的信息

2.1 查看系统硬件信息



lscpu [选项]

功能:

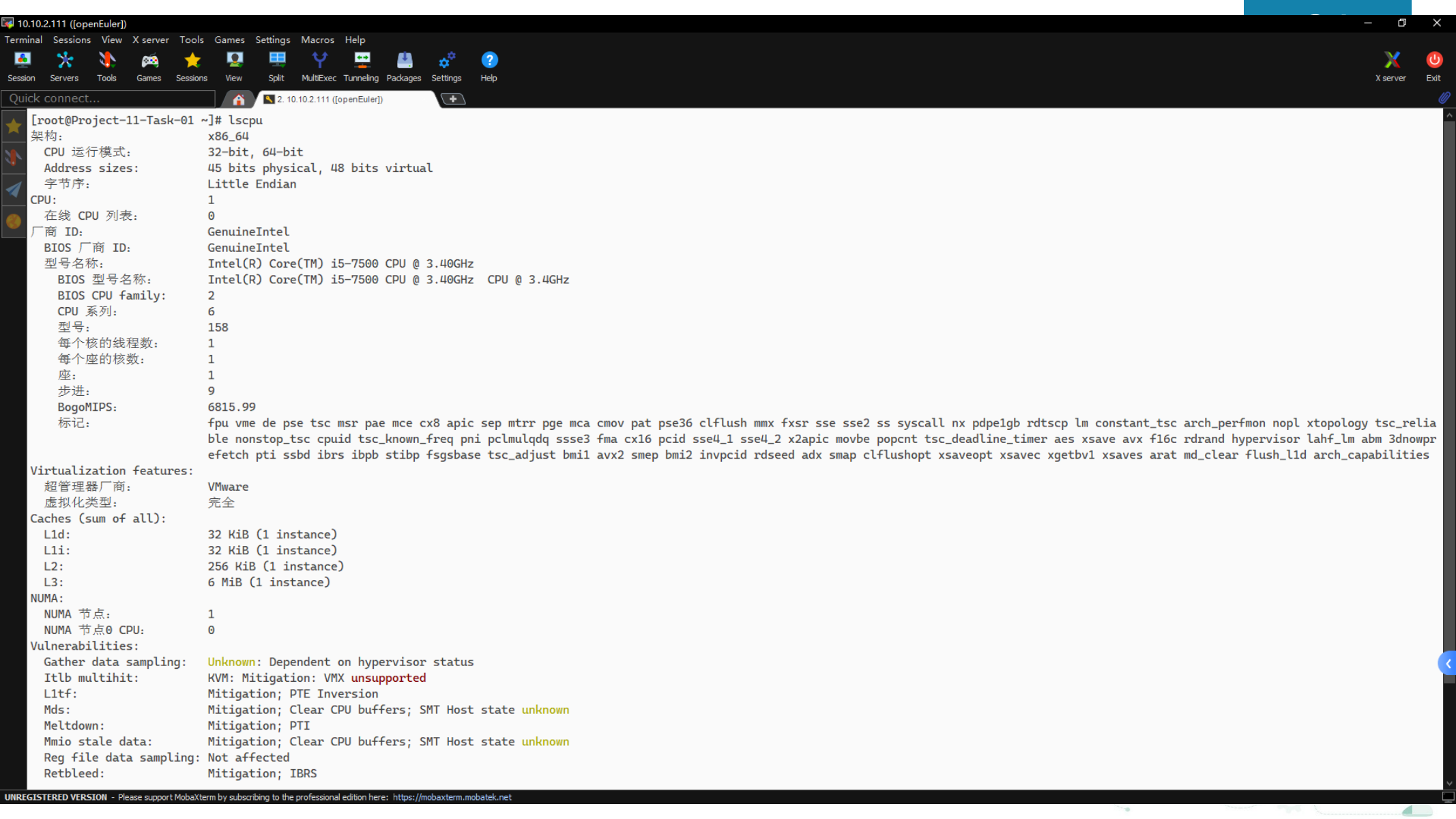
- 查看 CPU 信息。

参数/命令:

主要选项:

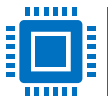
- -e: 以扩展可读的格式显示
- -p: 以可解析的格式显示





2. 查看本地主机的信息

2.1 查看系统硬件信息



top [选项]

功能:

- 实时监控 CPU 的使用情况。

参数/命令:

主要选项:

- b: 以批处理模式操作
- c: 显示整个命令行
- d: 屏幕刷新闻隔时间
- l: 忽略失效过程
- s: 保密模式
- S: 累积模式
- I <时间>: 设置间隔时间
- u <用户名>: 指定用户名
- p <进程号>: 指定进程号
- n <次数>: 循环次数



```

10.10.2.111 (openEuler)
Terminal Sessions View X server Tools Games Settings Macros Help
Session Servers Tools Games Sessions View Split MultiExec Tunneling Packages Settings Help
Quick connect... 2. 10.10.2.111 (openEuler)
top - 18:03:00 up 7:18, 1 user, load average: 0.00, 0.00, 0.00
Tasks: 184 total, 1 running, 183 sleeping, 0 stopped, 0 zombie
%Cpu(s): 0.3 us, 0.3 sy, 0.0 ni, 99.4 id, 0.0 wa, 0.0 hi, 0.0 si, 0.0 st
MiB Mem : 930.0 total, 388.7 free, 384.4 used, 297.6 buff/cache
MiB Swap: 2048.0 total, 2048.0 free, 0.0 used, 545.6 avail Mem

  PID USER      PR  NI  VIRT  RES  SHR  S  %CPU  %MEM    TIME+  COMMAND
 1846 root        20   0     0     0     0   I   0.3   0.0   0:03.10 kworker/0:0-mpt_poll_0
 1928 root        20   0 26916 5908 3672  R   0.3   0.6   0:00.02 top
    1 root        20   0 71540 26008 9408  S   0.0   2.7   0:01.28 systemd
    2 root        20   0     0     0     0   S   0.0   0.0   0:00.00 kthreadd
    3 root        20   0     0     0     0   S   0.0   0.0   0:00.00 pool_workqueue_release
    4 root         0 -20     0     0     0   I   0.0   0.0   0:00.00 kworker/R-rcu_g
    5 root         0 -20     0     0     0   I   0.0   0.0   0:00.00 kworker/R-rcu_p
    6 root         0 -20     0     0     0   I   0.0   0.0   0:00.00 kworker/R-slub_
    7 root         0 -20     0     0     0   I   0.0   0.0   0:00.00 kworker/R-nets
    9 root         0 -20     0     0     0   I   0.0   0.0   0:00.00 kworker/0:0H-events_highpri
   11 root        20   0     0     0     0   I   0.0   0.0   0:00.59 kworker/u256:0-ext4-rsv-conver
   12 root         0 -20     0     0     0   I   0.0   0.0   0:00.00 kworker/R-mm_pe
   13 root        20   0     0     0     0   I   0.0   0.0   0:00.00 rcu_tasks_rude_kthread
   14 root        20   0     0     0     0   I   0.0   0.0   0:00.00 rcu_tasks_trace_kthread
   15 root        20   0     0     0     0   S   0.0   0.0   0:00.04 ksoftirqd/0
   16 root        20   0     0     0     0   I   0.0   0.0   0:00.10 rcu_sched
   17 root        rt    0     0     0     0   S   0.0   0.0   0:00.03 migration/0
   18 root       -51   0     0     0     0   S   0.0   0.0   0:00.00 idle_inject/0
   19 root        20   0     0     0     0   S   0.0   0.0   0:00.00 cpuhp/0
   21 root        20   0     0     0     0   S   0.0   0.0   0:00.00 kdevtmpfs
   22 root         0 -20     0     0     0   I   0.0   0.0   0:00.00 kworker/R-inet_
   23 root        20   0     0     0     0   S   0.0   0.0   0:00.00 kauditd
   25 root        20   0     0     0     0   S   0.0   0.0   0:00.01 khungtaskd
   26 root        20   0     0     0     0   S   0.0   0.0   0:00.00 oom_reaper
   29 root         0 -20     0     0     0   I   0.0   0.0   0:00.00 kworker/R-write
   30 root        20   0     0     0     0   S   0.0   0.0   0:00.30 kcompactd0
   31 root        25   5     0     0     0   S   0.0   0.0   0:00.00 ksmd
   32 root        39  19     0     0     0   S   0.0   0.0   0:00.27 khugepaged
   33 root         0 -20     0     0     0   I   0.0   0.0   0:00.00 kworker/R-crypt
   34 root         0 -20     0     0     0   I   0.0   0.0   0:00.00 kworker/R-kinte
   35 root         0 -20     0     0     0   I   0.0   0.0   0:00.00 kworker/R-kbloc
   36 root         0 -20     0     0     0   I   0.0   0.0   0:00.00 kworker/R-blkcg
   37 root         0 -20     0     0     0   I   0.0   0.0   0:00.00 kworker/R-tpm_d
   38 root         0 -20     0     0     0   I   0.0   0.0   0:00.00 kworker/R-md
   39 root         0 -20     0     0     0   I   0.0   0.0   0:00.00 kworker/R-md_bi

```

在总览运行信息呈现中，“load average”共有3个值，分别代表

- 前1分钟平均CPU负载、前5分钟平均CPU负载、前15分钟平均CPU负载。
- 在CPU运行信息呈现中，每个字段所表示的具体含义如下。

- us: 用户使用CPU的百分比
- sy: 内核使用CPU的百分比
- ni: 进程优先级改变使用CPU的百分比
- id: 空闲CPU的百分比
- wa: I/O等待使用CPU的百分比
- hi: 硬件中断使用CPU的百分比
- si: 软件中断使用CPU的百分比
- st: 系统实时使用CPU的百分比

在执行top命令时，可使用交互命令进行快捷操作。

- k: 终止一个进程
- i: 忽略闲置和僵死进程
- q: 退出程序
- r: 重新设置一个进程的优先级别
- S: 切换到累积模式
- s: 改变刷新时间（单位为s），输入0值则不断刷新，默认值是5s
- f或F: 从当前显示中添加或删除项目
- o或O: 改变显示项目的顺序
- I: 切换显示平均负载和启动时间信息
- m: 切换显示内存信息
- t: 切换显示进程和CPU状态信息
- c: 切换显示命令名称和完整命令行
- M: 根据驻留内存大小进行排序
- P: 根据CPU使用百分比大小进行排序
- T: 根据时间/累积时间进行排序

2. 查看本地主机的信息

2.1 查看系统硬件信息



htop [选项]

功能:

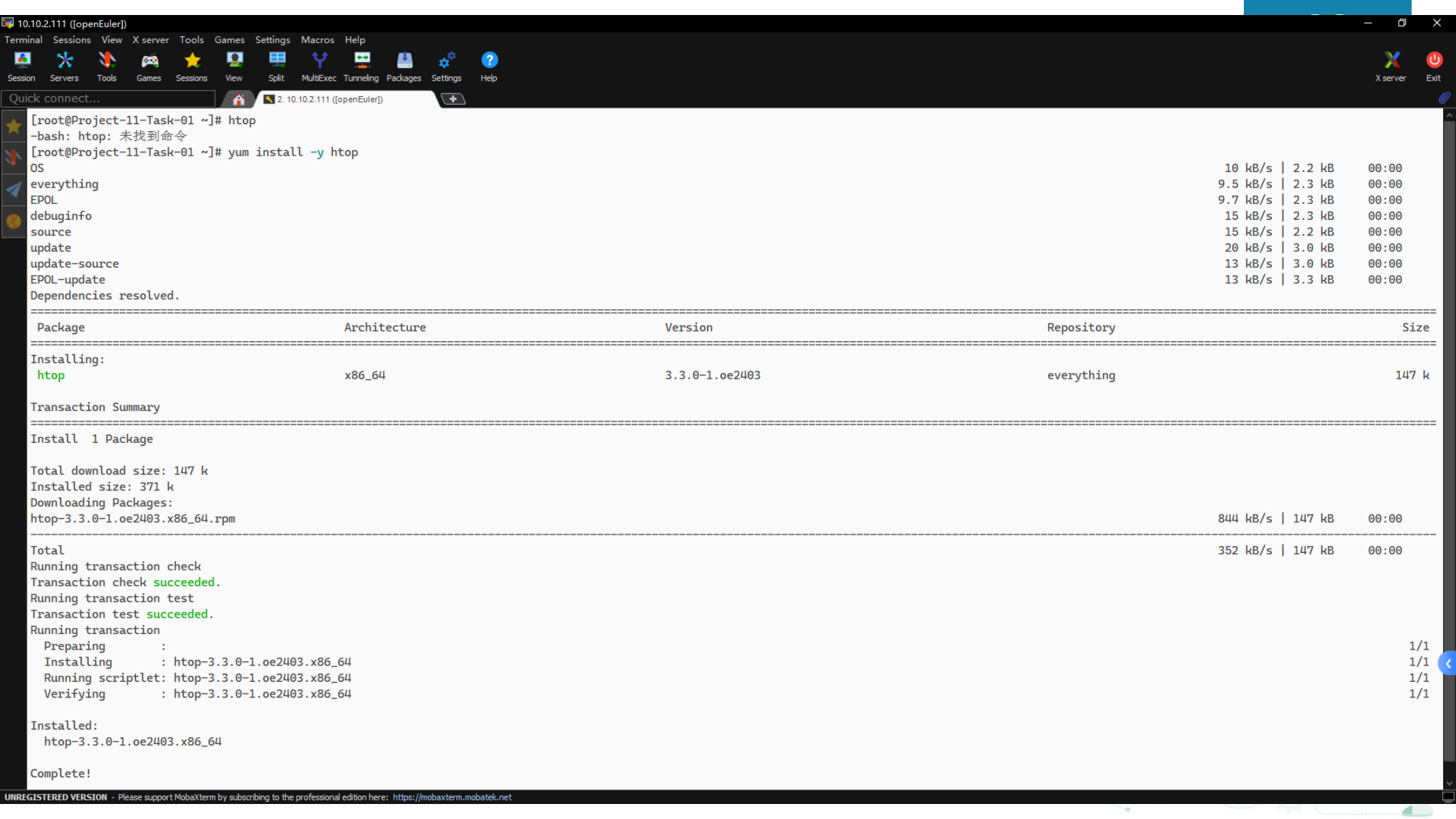
- 实时监控 CPU 的使用情况。
- 是具有操作互动的监控查看器，需要安装。

参数/命令:

主要选项:

- -C 或--no-color:
 - 使用一个单色的配色方案
- -d 或--delay=DELAY:
 - 设置更新时间，单位 s
- -u 或--user=USERNAME:
 - 只显示一个给定的用户的进程
- -p 或--pid= PID,[,PID,PID...]:
 - 只显示给定的 PIDs（进程号组信息）
- -s 或--sort-key COLUMN:
 - 以给定的列进行排序





```
[root@Project-11-Task-01 ~]# htop
-bash: htop: 未找到命令
[root@Project-11-Task-01 ~]# yum install -y htop
OS
everything
EPOL
debuginfo
source
update
update-source
EPOL-update
Dependencies resolved.
```

```
10 kB/s | 2.2 kB 00:00
9.5 kB/s | 2.3 kB 00:00
9.7 kB/s | 2.3 kB 00:00
15 kB/s | 2.3 kB 00:00
15 kB/s | 2.2 kB 00:00
20 kB/s | 3.0 kB 00:00
13 kB/s | 3.0 kB 00:00
13 kB/s | 3.3 kB 00:00
```

Package	Architecture	Version	Repository	Size
Installing: htop	x86_64	3.3.0-1.oe2403	everything	147 k

Transaction Summary

Install 1 Package

Total download size: 147 k
 Installed size: 371 k
 Downloading Packages:
 htop-3.3.0-1.oe2403.x86_64.rpm

```
844 kB/s | 147 kB 00:00
-----
352 kB/s | 147 kB 00:00
```

Total
 Running transaction check
 Transaction check succeeded.
 Running transaction test
 Transaction test succeeded.

```
Running transaction
  Preparing      :
  Installing     : htop-3.3.0-1.oe2403.x86_64 1/1
  Running scriptlet: htop-3.3.0-1.oe2403.x86_64 1/1
  Verifying     : htop-3.3.0-1.oe2403.x86_64 1/1
```

Installed:
 htop-3.3.0-1.oe2403.x86_64

Complete!

10.10.2.111 (openEuler)

Terminal Sessions View X server Tools Games Settings Macros Help

Session Servers Tools Games Sessions View Split MultiExec Tunneling Packages Settings Help

Quick connect... 2. 10.10.2.111 (openEuler)

```

CPU[]
Mem[|||||]
Swp[]

0.7%] Tasks: 24, 16 thr, 162 kthr; 0 running
245M/930M] Load average: 0.00 0.00 0.00
0K/2.00G] Uptime: 07:21:49
  
```

Main		I/O																			
PID	USER	PRI	NI	VIRT	RES	SHR	S	CPU%	MEM%	TIME+	Command										
1	root	20	0	71672	26024	9408	S	0.0	2.7	0:01.35	/usr/lib/systemd/systemd --switched-root --system --deserialize=34										
742	root	20	0	18532	8372	7248	S	0.0	0.9	0:00.15	/usr/lib/systemd/systemd-journald										
791	root	20	0	34068	12136	7252	S	0.0	1.3	0:00.17	/usr/lib/systemd/systemd-udev										
870	root	16	-4	18928	2604	1848	S	0.0	0.3	0:00.02	/sbin/auditd										
871	root	16	-4	18928	2604	1848	S	0.0	0.3	0:00.00	/sbin/auditd										
893	dbus	20	0	8652	4976	4164	S	0.0	0.5	0:00.07	/usr/bin/dbus-daemon --system --address=systemd: --nofork --nopidfile --systemd-activation --syslog-only										
900	root	20	0	237M	11472	9700	S	0.0	1.2	0:00.13	/usr/sbin/rngd -f -O jitter:timeout:10										
901	root	20	0	23040	13908	6936	S	0.0	1.5	0:00.10	/usr/lib/systemd/systemd-logind										
911	root	20	0	148M	45156	19260	S	0.0	4.7	0:00.47	/usr/bin/python3 -s /usr/sbin/firewalld --nofork --nopid										
922	root	20	0	62244	29336	1416	S	0.0	3.1	0:00.00	/usr/sbin/restorecond										
928	chrony	20	0	85028	3744	3112	S	0.0	0.4	0:00.08	/usr/sbin/chronyd										
933	root	20	0	237M	11472	9700	S	0.0	1.2	0:07.83	/usr/sbin/rngd -f -O jitter:timeout:10										
941	root	20	0	339M	19076	16508	S	0.0	2.0	0:00.10	/usr/sbin/NetworkManager --no-daemon										
955	root	20	0	339M	19076	16508	S	0.0	2.0	0:00.34	/usr/sbin/NetworkManager --no-daemon										
956	root	20	0	339M	19076	16508	S	0.0	2.0	0:00.00	/usr/sbin/NetworkManager --no-daemon										
957	root	20	0	339M	19076	16508	S	0.0	2.0	0:00.00	/usr/sbin/NetworkManager --no-daemon										
972	root	20	0	15440	8508	7260	S	0.0	0.9	0:00.02	sshd: /usr/sbin/sshd -D [listener] 0 of 10-100 startups										
977	root	20	0	277M	30792	15816	S	0.0	3.2	0:02.05	/usr/bin/python3 -Es /usr/sbin/tuned -l -P										
992	root	20	0	23228	3404	2552	S	0.0	0.4	0:00.07	/usr/sbin/crond -n										
995	root	20	0	21548	1908	1764	S	0.0	0.2	0:00.01	/sbin/agetty -o -p -- \u --noclear - linux										
1064	polkitd	20	0	300M	8292	7020	S	0.0	0.9	0:00.07	/usr/lib/polkit-1/polkitd --no-debug										
1097	polkitd	20	0	300M	8292	7020	S	0.0	0.9	0:00.00	/usr/lib/polkit-1/polkitd --no-debug										
1099	polkitd	20	0	300M	8292	7020	S	0.0	0.9	0:00.00	/usr/lib/polkit-1/polkitd --no-debug										
1100	polkitd	20	0	300M	8292	7020	S	0.0	0.9	0:00.00	/usr/lib/polkit-1/polkitd --no-debug										
1126	root	20	0	148M	45156	19260	S	0.0	4.7	0:00.00	/usr/bin/python3 -s /usr/sbin/firewalld --nofork --nopid										
1151	root	20	0	156M	6460	5020	S	0.0	0.7	0:00.00	/usr/sbin/rsyslogd -n -i/var/run/rsyslogd.pid										
1176	root	20	0	156M	6460	5020	S	0.0	0.7	0:01.55	/usr/sbin/rsyslogd -n -i/var/run/rsyslogd.pid										
1179	root	20	0	156M	6460	5020	S	0.0	0.7	0:00.00	/usr/sbin/rsyslogd -n -i/var/run/rsyslogd.pid										
1337	root	20	0	277M	30792	15816	S	0.0	3.2	0:01.63	/usr/bin/python3 -Es /usr/sbin/tuned -l -P										
1342	root	20	0	277M	30792	15816	S	0.0	3.2	0:00.00	/usr/bin/python3 -Es /usr/sbin/tuned -l -P										
1343	root	20	0	277M	30792	15816	S	0.0	3.2	0:00.00	/usr/bin/python3 -Es /usr/sbin/tuned -l -P										
1348	root	20	0	237M	11472	9700	S	0.0	1.2	0:00.00	/usr/sbin/rngd -f -O jitter:timeout:10										
1349	root	20	0	237M	11472	9700	S	0.0	1.2	0:00.00	/usr/sbin/rngd -f -O jitter:timeout:10										
1371	root	20	0	17956	10680	9020	S	0.0	1.1	0:00.04	sshd: root [priv]										

F1 Help F2 Setup F3 Search F4 Filter F5 Tree F6 SortBy F7 Nice F8 Nice F9 Kill F10 Quit

UNREGISTERED VERSION - Please support MobaXterm by subscribing to the professional edition here: <https://mobaxterm.mobatek.net>

2. 查看本地主机的信息

- 操作系统中 CPU 相关的概念。

表 11-2-1 CPU 的基本概念及其含义

概念	说明
物理 CPU	主板上实际接入的 CPU 个数
CPU 核数	每个物理 CPU 上实际接入的芯片组数量，如双核、四核等
逻辑 CPU	一般情况下，逻辑 CPU 数=物理 CPU 数量×CPU 核数，如果逻辑 CPU 多于物理 CPU，说明该 CPU 支持超线程技术



2. 查看本地主机的信息

- CPU 的缓存分为 3 个级别：
 - L1、L2、L3，级别越小越接近 CPU 处理器，速度越快，容量越小。

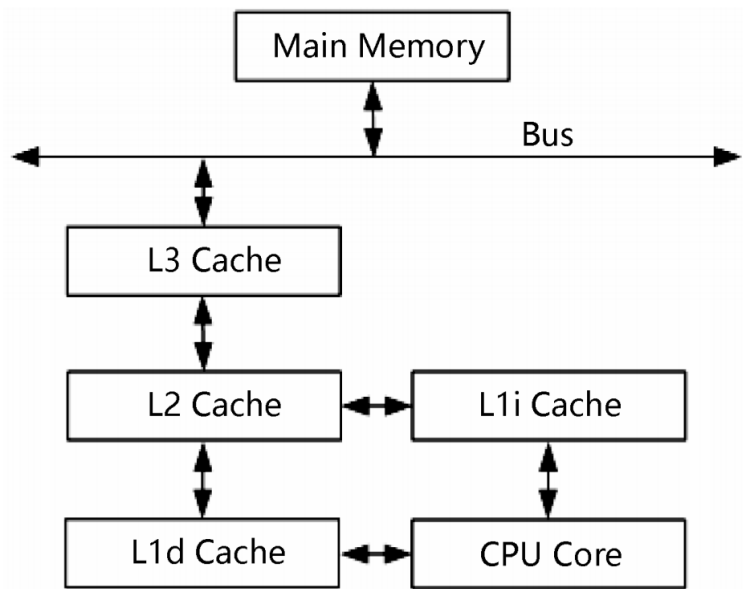


图 11-2-2 CPU 缓存结构



2. 查看本地主机的信息

2.1 查看系统硬件信息

- CPU负载表示系统上正在运行或等待运行的任务数量。
 - 负载可以使用负载平均值来衡量，通常是一段时间内的平均值，如 1min、5min、15min 的负载平均值。
 - 负载平均值是运行队列中的平均任务数量，包括正在执行和等待执行的任务。



2. 查看本地主机的信息

□ CPU 监控工具

■ top

- 能够实时显示系统中各个进程的资源占用状况（比如 CPU、内存以及进程的使用），默认 5s 刷新一下进程列表，类似于 Windows 的任务管理器。该工具集成在 openEuler 操作系统安装软件中，随系统一并安装。

■ mpstat

- mpstat（全称为 multiprocessor state）可以查看所有 CPU 的平均负载，也可以查看指定 CPU 的负载，是一款常用的多核 CPU 性能分析工具，用来实时监控 CPU 的性能指标。该工具集成在 sysstat 软件中，可使用 yum 工具安装。

■ vmstat

- vmstat 是最常见的 Linux/Unix 监控工具，可以呈现一定时间间隔的服务器状态值，包括服务器的 CPU 使用率、内存的使用情况、虚拟内存的交换情况、IO 读写情况等。该工具集成在 sysstat 软件中，可使用 yum 工具安装。



2. 查看本地主机的信息

□ CPU 监控工具

■ pidstat

- pidstat 工具用来监控 Linux 内核管理的独立进程，可查看每个进程的 CPU 使用情况。该工具集成在 sysstat 软件中，可使用 yum 工具安装。

■ dstat

- dstat 是一个全能系统信息统计工具，支持即时刷新，也可以收集指定的性能资源（如 dstat -c 即显示 CPU 的使用情况）。该工具为单独软件，可使用 yum 工具安装。

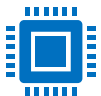
■ nmon

- nmon 工具能够动态地展示 openEuler 的多项性能，也可手动输入命令单项查看 CPU 性能。其支持获取 openEuler 性能数据，并通过 nmon_analyser 图形化工具进行分析后呈现其运行状态。该工具为单独软件，可使用 yum 工具安装。



2. 查看本地主机的信息

2.2 查看系统存储使用情况



free [选项]

功能:

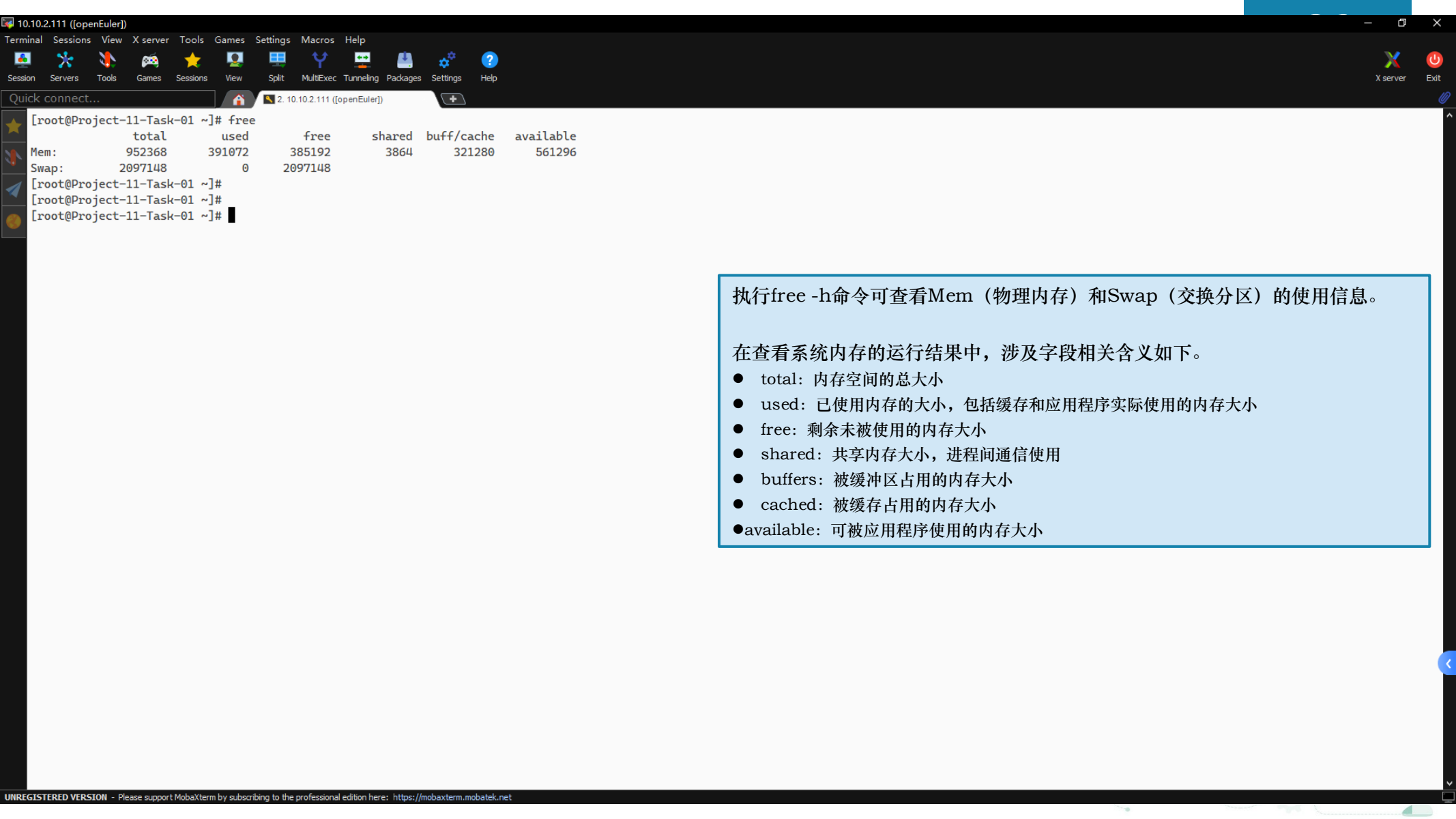
- 查看系统内存。

参数/命令:

主要选项:

- -b:
 - 以 Byte 为单位显示内存的使用情况
- -k:
 - 以 KB 为单位显示内存的使用情况
- -m:
 - 以 MB 为单位显示内存的使用情况
- -o:
 - 不显示缓冲区调节列
- -s <间隔秒数>:
 - 持续观察内存使用状况，按照指定时间刷新数据
- -t:
 - 显示内存总和列





执行free -h命令可查看Mem（物理内存）和Swap（交换分区）的使用信息。

在查看系统内存的运行结果中，涉及字段相关含义如下。

- total: 内存空间的总大小
- used: 已使用内存的大小，包括缓存和应用程序实际使用的内存大小
- free: 剩余未被使用的内存大小
- shared: 共享内存大小，进程间通信使用
- buffers: 被缓冲区占用的内存大小
- cached: 被缓存占用的内存大小
- available: 可被应用程序使用的内存大小

2. 查看本地主机的信息

2.2 查看系统存储使用情况



lsblk [选项] [设备]

功能:

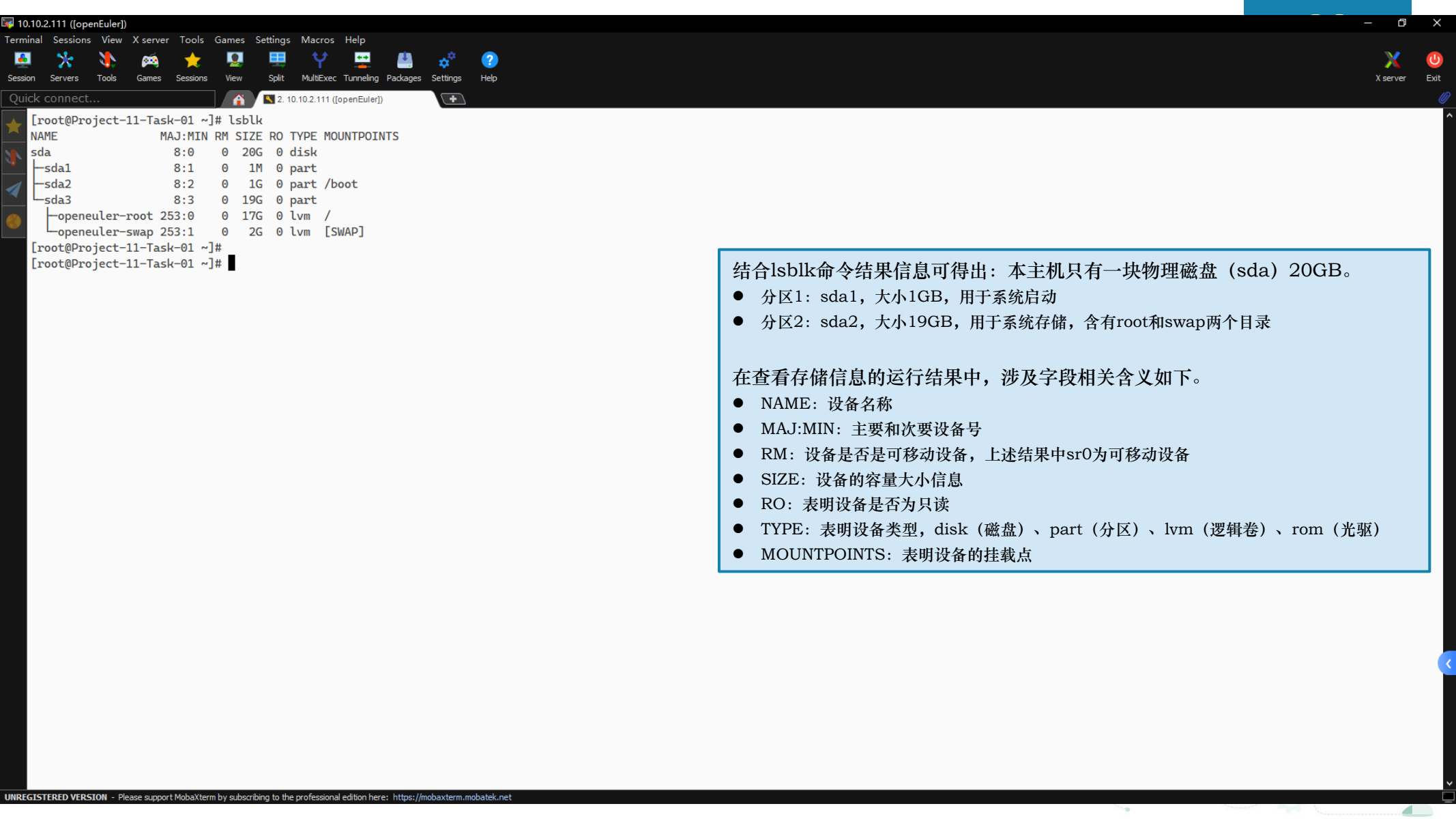
- 查看磁盘的相关信息以及磁盘分区分布情况等信
息。

参数/命令:

主要选项:

- -a: 打印所有设备
- -b: 以字节的形式输出
- -m: 输出磁盘分区的存储权限信息
- -S: 输出有关 SCSI 设备的信息
- -n: 不输出标题信息
- -l: 使用列表格式输出
- -d: 不输出从属关系的分区信息





结合lsblk命令结果信息可得出：本主机只有一块物理磁盘（sda）20GB。

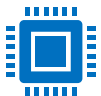
- 分区1：sda1，大小1GB，用于系统启动
- 分区2：sda2，大小19GB，用于系统存储，含有root和swap两个目录

在查看存储信息的运行结果中，涉及字段相关含义如下。

- NAME：设备名称
- MAJ:MIN：主要和次要设备号
- RM：设备是否是可移动设备，上述结果中sr0为可移动设备
- SIZE：设备的容量大小信息
- RO：表明设备是否为只读
- TYPE：表明设备类型，disk（磁盘）、part（分区）、lvm（逻辑卷）、rom（光驱）
- MOUNTPOINTS：表明设备的挂载点

2. 查看本地主机的信息

2.2 查看系统存储使用情况



iotop [选项]

功能:

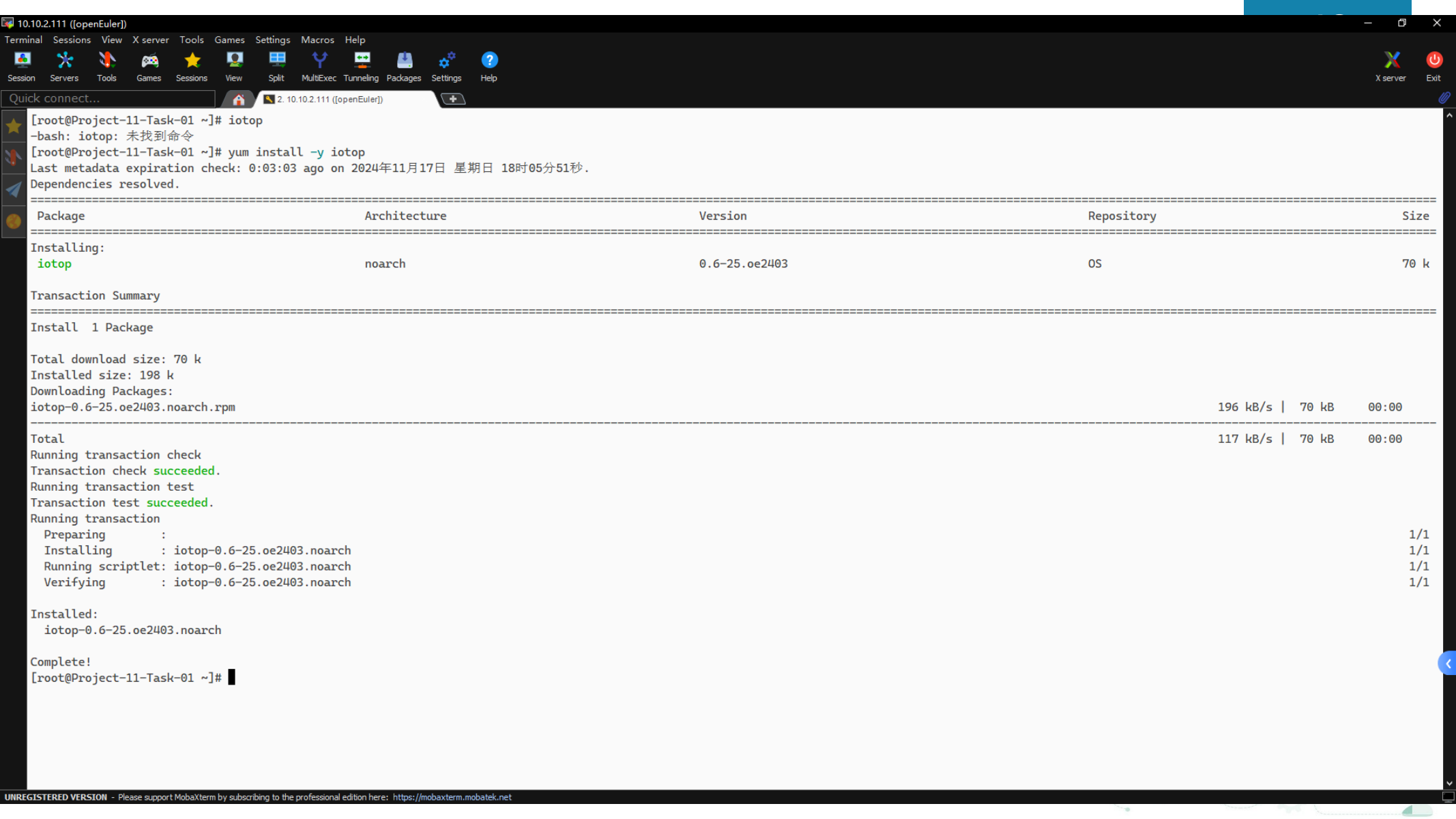
- 可监控磁盘 IO 的使用状况，并将关联的进程、用户等相关信息也一并输出，相当于从进程层面监控磁盘 IO。

参数/命令:

主要选项:

- -o: 只显示有 IO 操作的进程
- -b: 批量显示，无交互，主要用作记录到文件
- -n NUM:
 - 显示 NUM 次，主要用于非交互式模式
- -d SEC:
 - 间隔 SEC 秒显示一次
- -p PID:
 - 针对进程进行输出
- -u USER:
 - 根据进程执行用户进行输出





TID	PRIO	USER	DISK READ	DISK WRITE	SWAPIN	IO>	COMMAND
Total DISK READ :			0.00 B/s		Total DISK WRITE : 0.00 B/s		
Actual DISK READ:			0.00 B/s		Actual DISK WRITE: 0.00 B/s		
1	be/4	root	0.00 B/s	0.00 B/s	?unavailable?		systemd --switched-root --system --deserialize=34
2	be/4	root	0.00 B/s	0.00 B/s	?unavailable?		[kthreadd]
3	be/4	root	0.00 B/s	0.00 B/s	?unavailable?		[pool_workqueue_release]
4	be/0	root	0.00 B/s	0.00 B/s	?unavailable?		[kworker/R-rcu_g]
5	be/0	root	0.00 B/s	0.00 B/s	?unavailable?		[kworker/R-rcu_p]
6	be/0	root	0.00 B/s	0.00 B/s	?unavailable?		[kworker/R-slub_]
7	be/0	root	0.00 B/s	0.00 B/s	?unavailable?		[kworker/R-netns]
9	be/0	root	0.00 B/s	0.00 B/s	?unavailable?		[kworker/0:0H-events_highpri]
11	be/4	root	0.00 B/s	0.00 B/s	?unavailable?		[kworker/u256:0-ext4-rsv-conversion]
12	be/0	root	0.00 B/s	0.00 B/s	?unavailable?		[kworker/R-mm_pe]
13	be/4	root	0.00 B/s	0.00 B/s	?unavailable?		[rcu_tasks_rude_kthread]
14	be/4	root	0.00 B/s	0.00 B/s	?unavailable?		[rcu_tasks_trace_kthread]
15	be/4	root	0.00 B/s	0.00 B/s	?unavailable?		[ksoftirqd/0]
16	be/4	root	0.00 B/s	0.00 B/s	?unavailable?		[rcu_sched]
17	rt/4	root	0.00 B/s	0.00 B/s	?unavailable?		[migration/0]
18	rt/4	root	0.00 B/s	0.00 B/s	?unavailable?		[idle_inject/0]
19	be/4	root	0.00 B/s	0.00 B/s	?unavailable?		[cpuhp/0]
21	be/4	root	0.00 B/s	0.00 B/s	?unavailable?		[kdevtmpfs]
22	be/0	root	0.00 B/s	0.00 B/s	?unavailable?		[kworker/R-inet_]
23	be/4	root	0.00 B/s	0.00 B/s	?unavailable?		[kauditd]
25	be/4	root	0.00 B/s	0.00 B/s	?unavailable?		[khungtaskd]
26	be/4	root	0.00 B/s	0.00 B/s	?unavailable?		[oom_reaper]
29	be/0	root	0.00 B/s	0.00 B/s	?unavailable?		[kworker/R-wrrite]
30	be/4	root	0.00 B/s	0.00 B/s	?unavailable?		[kcompactd0]
31	be/5	root	0.00 B/s	0.00 B/s	?unavailable?		[ksmd]
32	be/7	root	0.00 B/s	0.00 B/s	?unavailable?		[khugepaged]
33	be/0	root	0.00 B/s	0.00 B/s	?unavailable?		[kworker/R-crypt]
34	be/0	root	0.00 B/s	0.00 B/s	?unavailable?		[kworker/R-kinte]
35	be/0	root	0.00 B/s	0.00 B/s	?unavailable?		[kworker/R-kbloc]
36	be/0	root	0.00 B/s	0.00 B/s	?unavailable?		[kworker/R-blkcg]
37	be/0	root	0.00 B/s	0.00 B/s	?unavailable?		[kworker/R-tpm_d]
38	be/0	root	0.00 B/s	0.00 B/s	?unavailable?		[kworker/R-md]
39	be/0	root	0.00 B/s	0.00 B/s	?unavailable?		[kworker/R-md_bi]
40	be/0	root	0.00 B/s	0.00 B/s	?unavailable?		[kworker/R-edac-]
41	rt/4	root	0.00 B/s	0.00 B/s	?unavailable?		[watchdogd]
43	be/4	root	0.00 B/s	0.00 B/s	?unavailable?		[kswapd0]
45	be/0	root	0.00 B/s	0.00 B/s	?unavailable?		[kworker/R-kthro]
CONFIG_TASK_DELAY_ACCT not enabled in kernel, cannot determine SWAPIN and IO %			49	rt/4	root	0.00 B/s	0.00 B/s ?unavailable? [irq/24-pciehp]
50	rt/4	root	0.00 B/s	0.00 B/s	?unavailable?		[irq/25-pciehp]

主机磁盘IO总览结果的字段如下。

- Total DISK READ: 每秒磁盘总读取大小
- Total DISK WRITE: 每秒磁盘总写入大小
- Actual DISK READ: 实际每秒磁盘读取大小
- Actual DISK WRITE: 实际每秒磁盘写入大小

主机磁盘IO详细信息结果的字段如下。

- TID: 线程ID
- PRIO: 线程优先级
- USER: 所属用户
- DISK READ: 每秒钟磁盘读取大小
- DISK WRITE: 每秒钟磁盘写入大小
- SWAPIN: 写入交换分区占比
- IO: IO使用率大小
- COMMAND: 线程执行命令

2. 查看本地主机的信息

2.2 查看系统存储使用情况

- 系统内存分为物理内存和虚拟内存两类。
- 物理内存。
 - 物理内存由半导体器件制成，是 CPU 能直接寻址的存储空间，具有存取速度快的特点。物理内存的作用。
 1. 暂时存放 CPU 的运算数据。
 2. 存储硬盘等外部存储器交换的数据。
 3. 保障 CPU 计算的稳定性和高性能。
- 虚拟内存。
 - 虚拟内存是操作系统为了解决物理内存不足而提出的策略，其利用磁盘空间虚拟出一块逻辑内存，用作虚拟内存的磁盘空间被称为交换空间（Swap Space）。
 - 作为物理内存的扩展，Linux 操作系统会在物理内存不足时，将暂时不用的内存块信息写到交换空间，从而释放部分物理内存，方便其他进程使用。当需要使用存储的内存信息时，内核将信息重新从交换空间读入物理内存中进行操作。



2. 查看本地主机的信息

2.2 查看系统存储使用情况

- 系统内存分为物理内存和虚拟内存两类。
- 虚拟内存。
 - 使用虚拟内存的主要优势。
 1. 获取更多的内存空间，且空间地址是连续的。
 2. 程序隔离。不同进程的虚拟地址之间没有关系，单个进程操作不会对其他进程造成影响。
 3. 数据保护。每块虚拟内存都有相应的读写属性，保护程序的代码段不被修改，数据块不能被执行等，增加了系统的安全性。
 4. 内存映射。可直接映射磁盘上的文件到虚拟地址空间，从而做到物理内存长时间分配，只需要在读取相应文件的时候，从虚拟内存加载到物理内存中。
 5. 共享内存。进程间的内存共享可通过映射同一块物理内存到不同虚拟内存空间来实现共享。
 6. 使用虚拟内存后，可方便使用交换空间和复制写入（copy on write, COW）等功能。



2. 查看本地主机的信息

2.2 查看系统存储使用情况

□ 内存工作机制。

- 在 Linux 操作系统中，以应用程序读写文件数据为例介绍内存的执行过程。
 1. 操作系统分配内存，将读取的数据从磁盘读入到内存中。
 2. 从内存中将数据分发给应用程序。
 3. 向文件中写数据时，操作系统分配内存接收用户数据。
 4. 接收完成后，内存将数据写入磁盘。
- 如果有大量数据需要从磁盘读取到内存或者由内存写入磁盘时，系统的读写性能就变得非常低下，因为无论是从磁盘读数据，还是写数据到磁盘，都是很消耗时间和资源的过程。



2. 查看本地主机的信息

2.2 查看系统存储使用情况

□ 文件系统

- 文件和目录的操作命令、存储、组织和控制的总体结构统称为文件系统。
- 文件系统是指格式化后用于存储文件的设备。
- 文件系统对存储空间进行组织和分配，并对文件的访问进行保护和控制。
- 不同的操作系统对文件的组织方式会有所区别，其所支持的文件系统类型也不一样。在Linux操作系统中，文件系统的组织方式是树状的层次式目录结构，在这个结构中处于最顶层的是根目录，用“/”代表，往下延伸是其各级子目录。

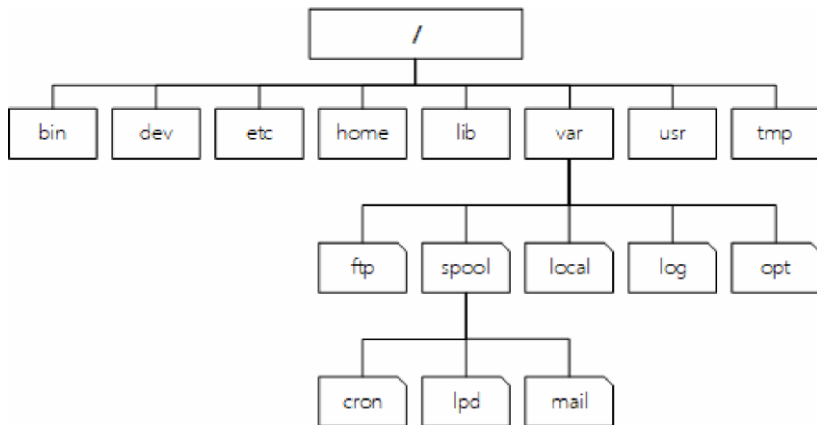


图 11-3-1 Linux 操作系统的文件系统结构示例



2. 查看本地主机的信息

2.2 查看系统存储使用情况

□ 磁盘 IO

- 磁盘 IO 是一种文件操作，用户进程产生的数据 IO 请求将通过 VFS、文件系统交给调度层进行排序和合并处理，再发送给块设备驱动进行最终数据读取和写入。

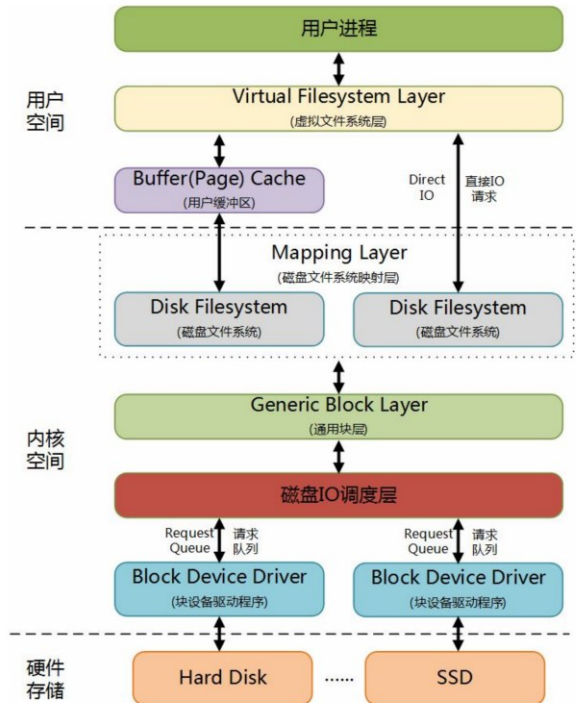


图 11-3-2 磁盘 IO 的工作模式



2. 查看本地主机的信息

2.2 查看系统存储使用情况

□ 磁盘IO的性能指标

■ 利用率

- 磁盘处理 IO 的时间百分比，磁盘 IO 利用率=(磁盘读取速度+磁盘写入速度)/(磁盘最大读取速度+磁盘最大写入速度)。如果过度使用通常意味着磁盘 IO 存在性能瓶颈。

■ 饱和度

- 磁盘处理 IO 的繁忙程度。过度饱和意味着磁盘存在严重的性能瓶颈。当饱和度为 100%时，磁盘无法接收新的 IO 请求。

■ IOPS:

- 每秒 IO 请求的数量，也可以理解为每秒钟磁盘进行多少次 IO 读写。

■ 吞吐量:

- 每秒磁盘 IO 请求数量的大小，也可以理解为每秒读写数据的总大小了。
- 吞吐量=IOPS*IO 大小。

■ 响应时间:

- 发送磁盘 IO 请求和接收响应之间的时间间隔。



2. 查看本地主机的信息

2.3 查看网络通信情况



ip neigh [选项] [参数]

功能:

- 查看 MAC 地址表。

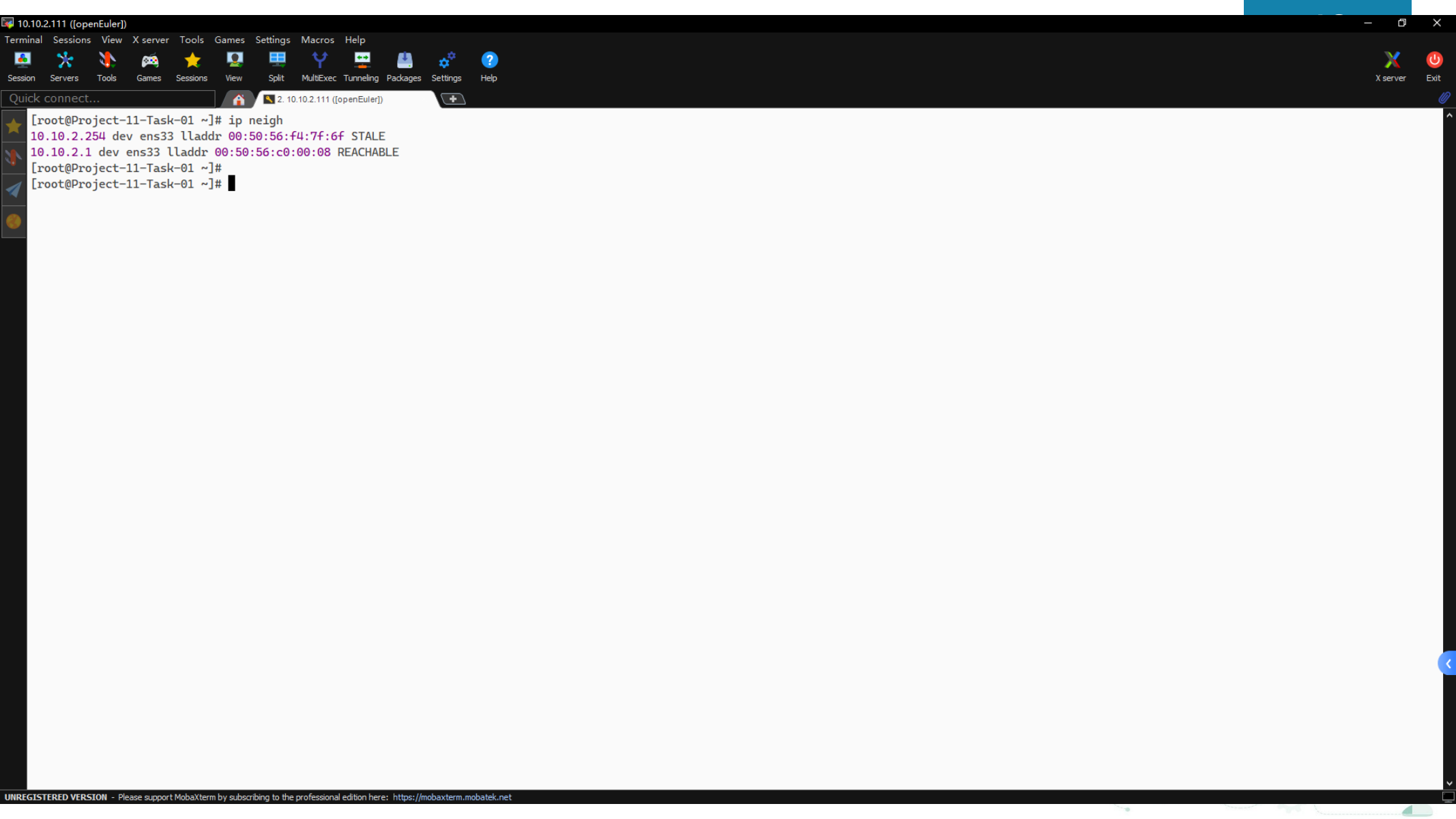
参数/命令:

- <ADDRESS(default)>
 - 记录目标主机的 IP 地址，可以是 IPv4 或 IPv6
- dev <NAME>
 - 对应连接的网卡设备接口
- lladdr <LLADDRESS>
 - 目标主机的链路地址 (MAC 地址)
- nud <NUD_STATE>
 - MAC 地址表状态
 - nud 是 Neighbour Unreachability Detection 的缩写

主要选项:

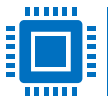
- add (a) :
 - 添加新的 MAC 地址表信息
- change (chg) :
 - 更改 MAC 地址表信息
- replace (repl) :
 - 替换 MAC 地址表信息
- delete (del/d) :
 - 删除 MAC 地址表信息
- flush (f) :
 - 刷新 MAC 地址表信息，无需具体参数
- show/list (sh/ls) :
 - 查看 MAC 地址表信息，可根据添加时的参数进行指定筛选查找





2. 查看本地主机的信息

2.3 查看网络通信情况



ss [选项]

功能:

- 显示处于活动状态的套接字信息，能够显示详细的网络会话连接状态信息。

参数/命令:

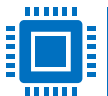
主要选项:

- -n: 不解析服务名称，以数字方式显示
- -a: 显示所有的套接字
- -l: 显示处于监听状态的套接字
- -o: 显示计时器信息
- -m: 显示套接字的内存使用情况
- -p: 显示使用套接字的进程信息
- -i: 显示内部的 TCP 信息
- -4: 显示 IPv4 套接字
- -6: 显示 IPv6 套接字
- -t: 显示 TCP 套接字
- -u: 显示 UDP 套接字
- -d: 显示 DCCP 套接字
- -w: 显示 RAW 套接字
- -x: 显示 UNIX 域套接字



2. 查看本地主机的信息

2.3 查看网络通信情况



iftop [选项]

功能:

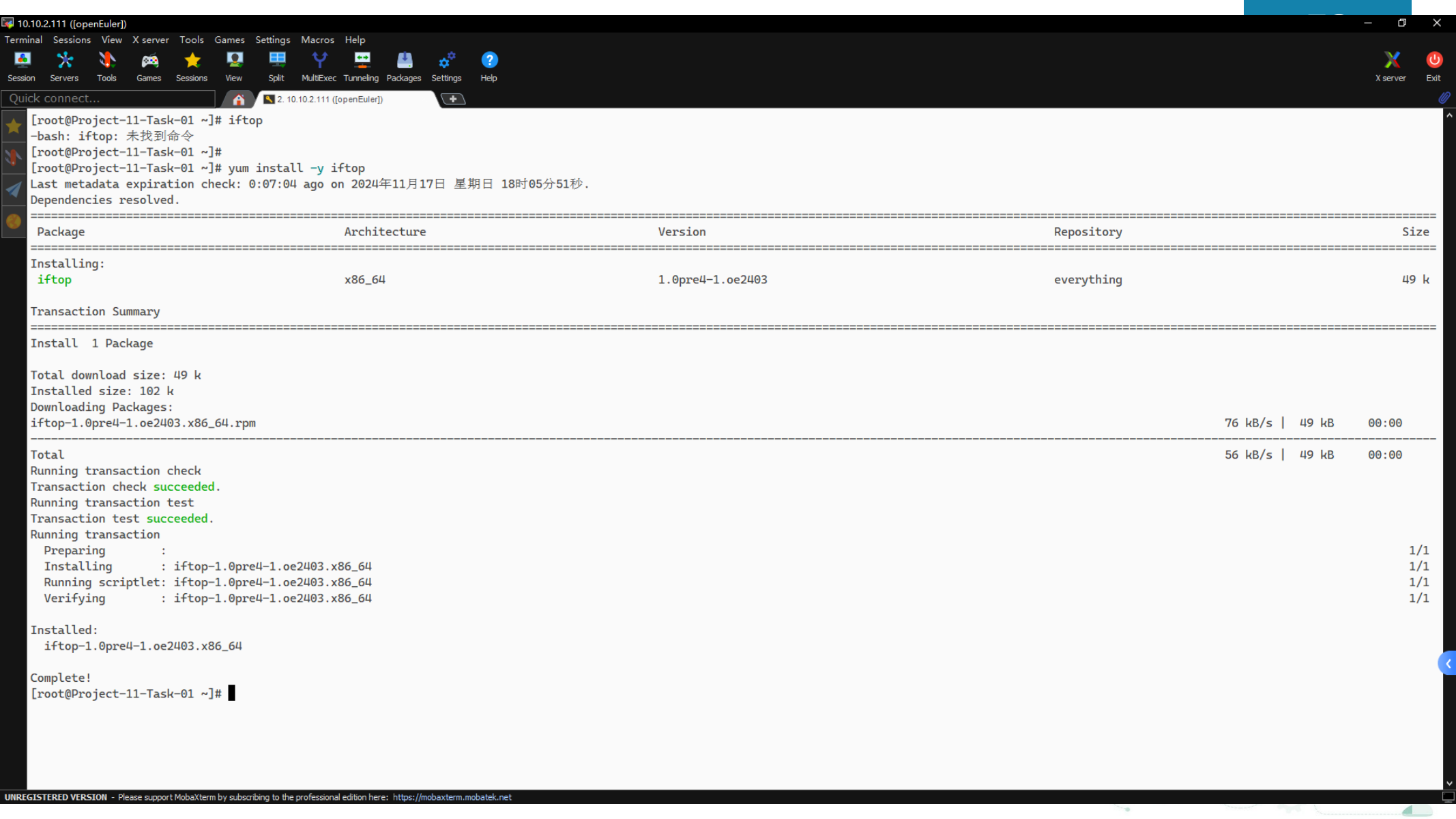
- 查看主机实时网络流量。

参数/命令:

主要选项:

- -i: 设定检测的网卡
- -B: 以 bytes 为单位显示流量 (默认是 bits)
- -n: 使 host 信息默认显示 IP
- -N: 使端口信息默认显示端口号
- -F: 显示特定网段的流入/流出流量大小
- -p: 运行混杂模式 (显示同一网段上其他主机的通信)
- -b: 显示流量图形条, 默认显示
- -f: 用于计算过滤包信息
- -P: 使 host 信息及端口信息默认均显示
- -m:
 - 设置界面最上边的刻度的最大值, 刻度分为 5 个大段显示





```
[root@Project-11-Task-01 ~]# iftop
-bash: iftop: 未找到命令
[root@Project-11-Task-01 ~]#
[root@Project-11-Task-01 ~]# yum install -y iftop
Last metadata expiration check: 0:07:04 ago on 2024年11月17日 星期日 18时05分51秒.
Dependencies resolved.
```

Package	Architecture	Version	Repository	Size
Installing: iftop	x86_64	1.0pre4-1.oe2403	everything	49 k

Transaction Summary

Install 1 Package

Total download size: 49 k
Installed size: 102 k
Downloading Packages:
iftop-1.0pre4-1.oe2403.x86_64.rpm

76 kB/s | 49 kB 00:00

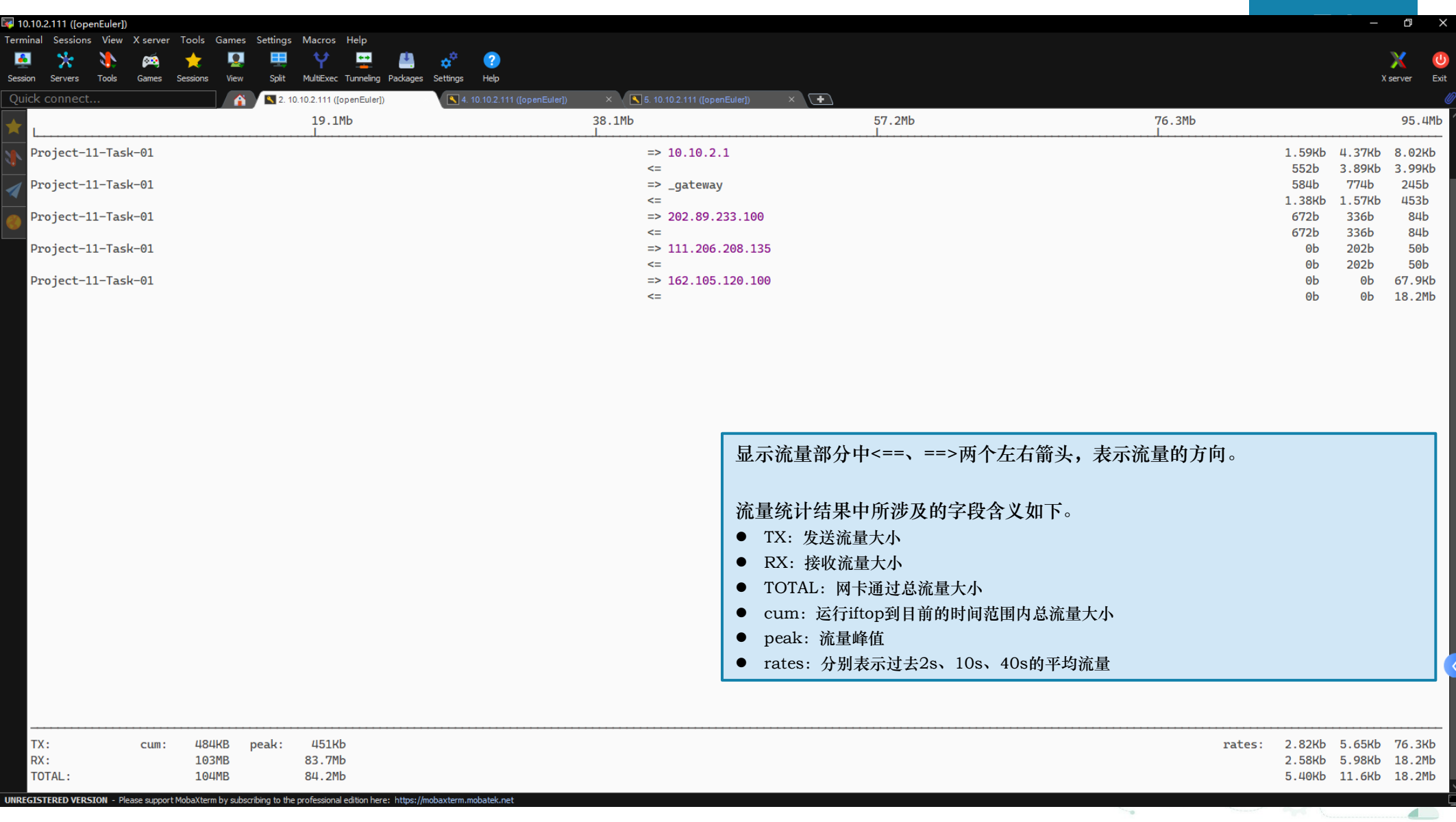
Total
Running transaction check
Transaction check succeeded.
Running transaction test
Transaction test succeeded.
Running transaction

56 kB/s | 49 kB 00:00

```
Preparing      :
Installing     : iftop-1.0pre4-1.oe2403.x86_64 1/1
Running scriptlet: iftop-1.0pre4-1.oe2403.x86_64 1/1
Verifying     : iftop-1.0pre4-1.oe2403.x86_64 1/1
```

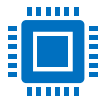
Installed:
iftop-1.0pre4-1.oe2403.x86_64

Complete!
[root@Project-11-Task-01 ~]#



2. 查看本地主机的信息

2.3 查看网络通信情况



ping [选项] [参数]

功能:

- 查看网络的连通性。

参数/命令:

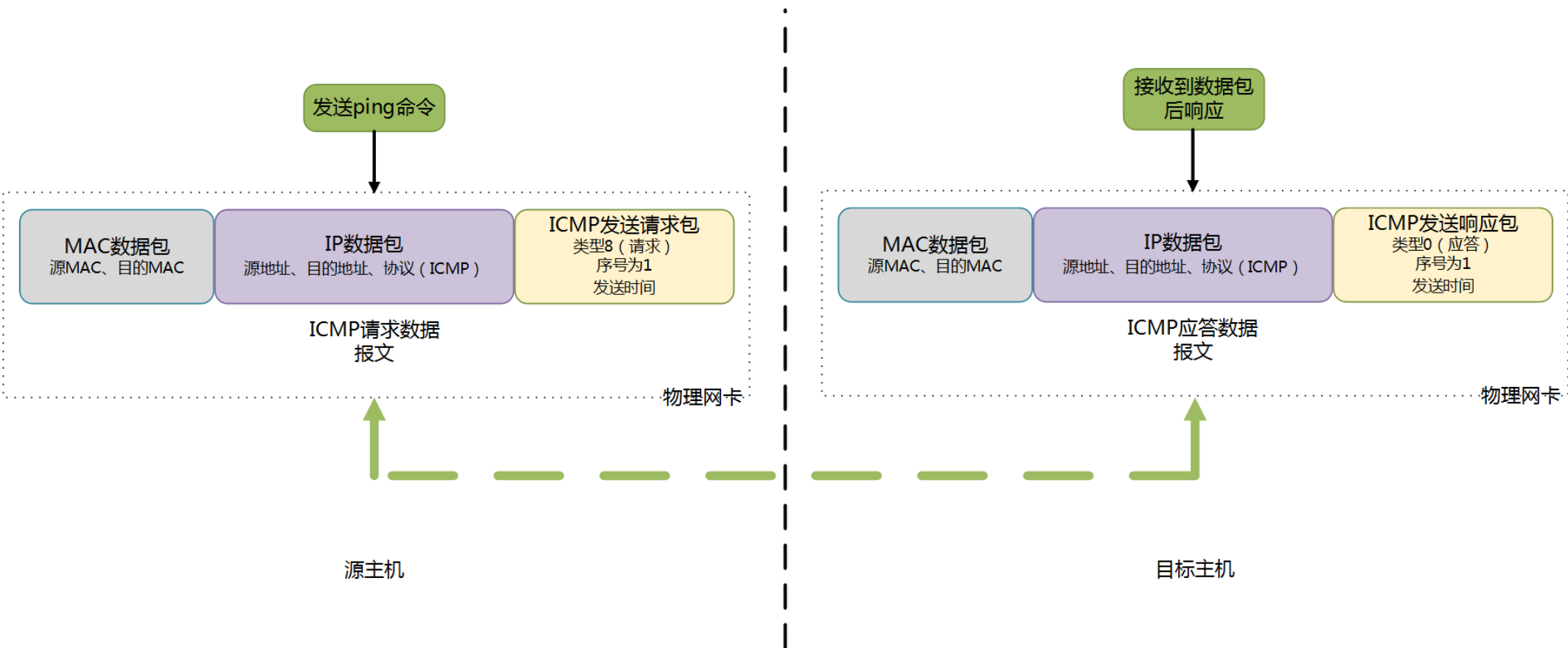
- hostname:
 - 指定检测的主机 IP 地址、主机名、域名

主要选项:

- -c <次数>:
 - 指定 ping 命令请求的次数
- -i <时间>:
 - 指定每个 ping 命令请求之间的时间间隔（以 s 为单位）
- -s <数据包大小>:
 - 指定发送的数据包大小（以 B 为单位），默认值为 56B
- -t <生存时间>:
 - 指定 ping 请求的生存周期最大时间，在 Linux 操作系统中默认最大为 128
- -q:
 - 只显示结果，不显示每个 ping 请求的详细信息
- -v:
 - 显示每个 ping 请求的详细信息

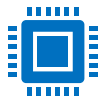


Ping命令的工作原理



2. 查看本地主机的信息

2.3 查看网络通信情况



traceroute [选项] [参数]

功能:

- 用于追踪数据包在网络上传输的全部节点及路径信息，能够及时追踪网络故障断点。

参数/命令:

- hostname:
 - 指定检测的主机 IP 地址、主机名、域名

主要选项:

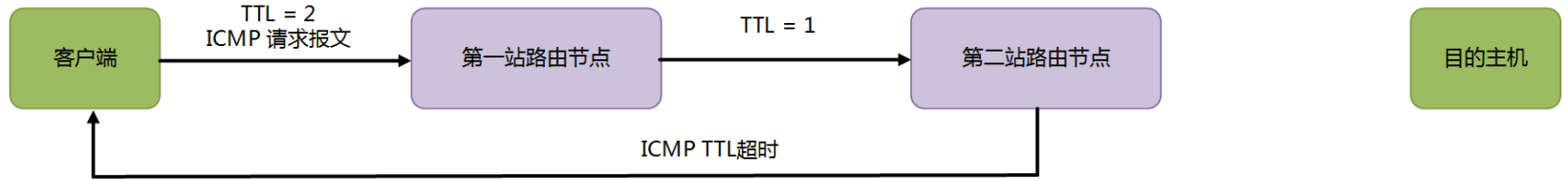
- -f <存活数值>:
 - 设置第一个检测数据包的存活数值 TTL 的大小
- -F: 不使用碎片数据报文
- -g <网关>: 设置来源路由网关，最多可设置 8 个
- -i <设备接口>: 使用指定的设备接口发送数据包
- -I: 使用 ICMP 取代 UDP 响应报文
- -m <存活数值>:
 - 设置检测数据包的最大存活数值 TTL 的大小
- -n: 直接使用 IP 地址而非主机名称
- -p <通信端口>: 指定传输协议的通信端口
- -r: 忽略普通路由表，直接将数据包送到远端主机上
- -s <来源地址>: 设置本地主机送出数据包的 IP 地址
- -t <服务类型>: 设置检测数据包的 TOS 数值
- -w <超时秒数>: 设置等待远端主机响应的时间
- -x: 开启或关闭数据包的正确性检验

Traceroute命令的工作原理

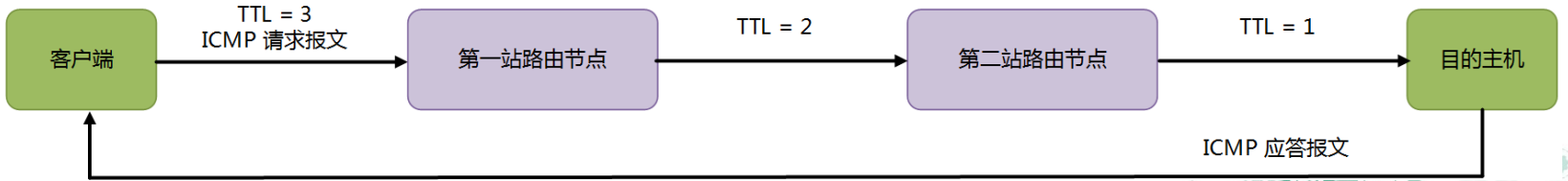
第①次路由
第一跳



第②次路由
第二跳



第③次路由
第三跳



2. 查看本地主机的信息

2.3 查看网络通信情况



mtr [选项] [参数]

功能：

- 用于主机网络的诊断与网络连通性的判断。

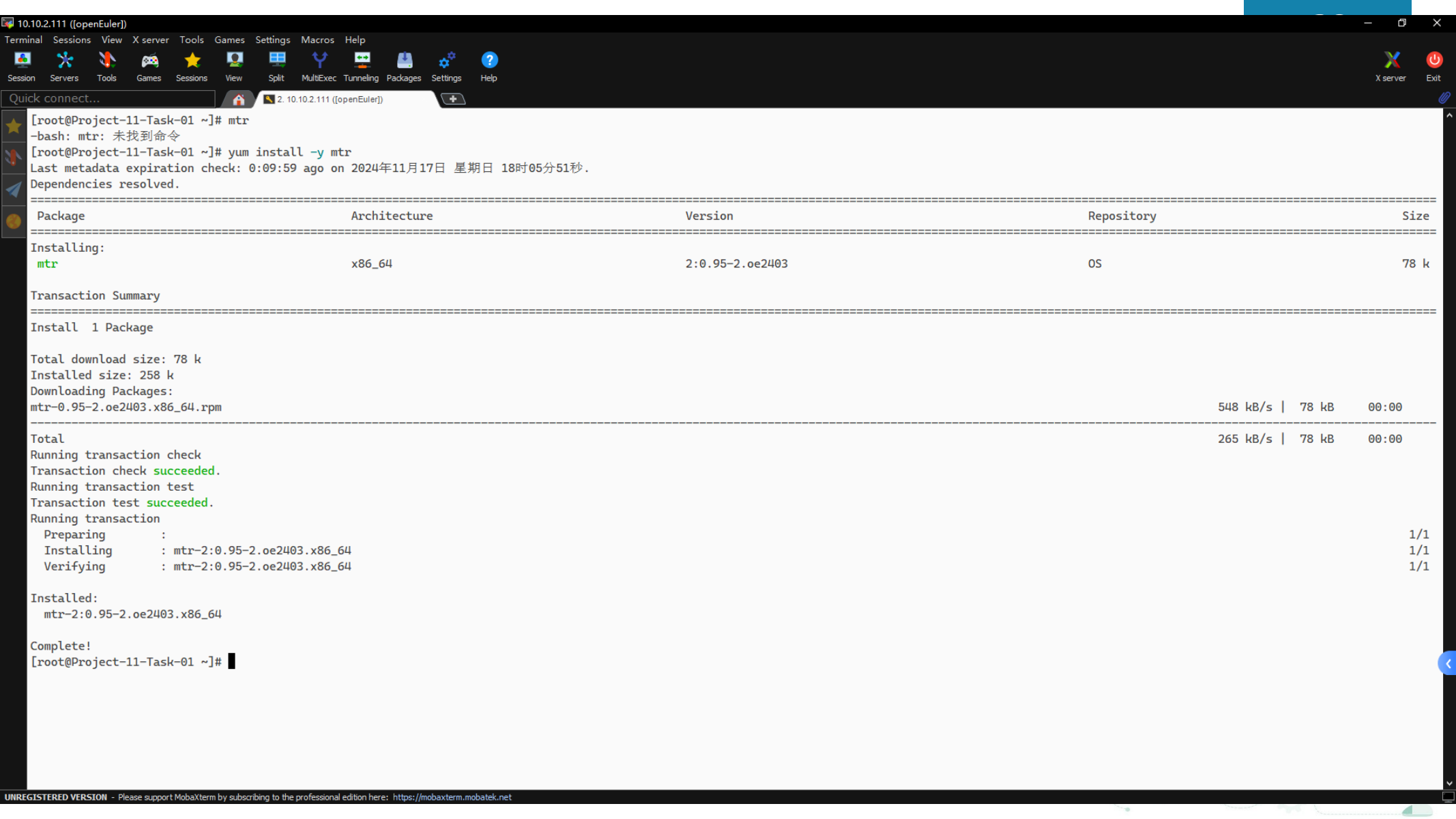
参数/命令：

- hostname:
 - 指定检测的主机 IP 地址、主机名、域名

主要选项：

- -r: 以报告模式显示
- -s: 指定 ping 数据包的大小
- --no-dns:
 - 不对 IP 地址做域名解析操作
- -a:
 - 设置发送数据包的 IP 地址（主机中设置多个 IP 地址）
- -i: 设置 ICMP 返回之间的时间，默认是 1s
- -4: 指定检测 IPv4 地址
- -6: 指定检测 IPv6 地址
- -c: 指定每秒发送数据包的数量





```
[root@Project-11-Task-01 ~]# mtr
```

```
-bash: mtr: 未找到命令
```

```
[root@Project-11-Task-01 ~]# yum install -y mtr
```

```
Last metadata expiration check: 0:09:59 ago on 2024年11月17日 星期日 18时05分51秒.
```

```
Dependencies resolved.
```

Package	Architecture	Version	Repository	Size
Installing:				
mtr	x86_64	2:0.95-2.oe2403	OS	78 k

Transaction Summary

Install 1 Package

Total download size: 78 k

Installed size: 258 k

Downloading Packages:

mtr-0.95-2.oe2403.x86_64.rpm 548 kB/s | 78 kB 00:00

Total 265 kB/s | 78 kB 00:00

Running transaction check

Transaction check **succeeded**.

Running transaction test

Transaction test **succeeded**.

Running transaction

Preparing	:			1/1
Installing	:	mtr-2:0.95-2.oe2403.x86_64		1/1
Verifying	:	mtr-2:0.95-2.oe2403.x86_64		1/1

Installed:
mtr-2:0.95-2.oe2403.x86_64

Complete!
[root@Project-11-Task-01 ~]#

```
10.10.2.111 (openEuler)
Terminal Sessions View X server Tools Games Settings Macros Help
Session Servers Tools Games Sessions View Split MultiExec Tunneling Packages Settings Help
Quick connect... 2. 10.10.2.111 (openEuler)
My traceroute [v0.95]
Project-11-Task-01 (10.10.2.111) -> www.edu.cn (202.205.109.203) 2024-11-17T18:23:45+0800
Keys: Help Display mode Restart statistics Order of fields quit

Host
1. _gateway
2. ipaddr203.cernet.cn

Packets
Loss% Snt Last Avg Best Wrst StDev
0.0% 8 0.4 0.4 0.3 0.6 0.1
0.0% 8 14.3 15.2 14.3 20.6 2.2
```

```
10.10.2.111 (openEuler)
Terminal Sessions View X server Tools Games Settings Macros Help
Session Servers Tools Games Sessions View Split MultiExec Tunneling Packages Settings Help
Quick connect... 2. 10.10.2.111 (openEuler)
My traceroute [v0.95]
Project-11-Task-01 (10.10.2.111) -> www.edu.cn (202.205.109.212) 2024-11-17T18:24:26+0800
Keys: Help Display mode Restart statistics Order of fields quit

Host
1. _gateway
2. (waiting for reply)

Packets
Loss% Snt Last Avg Best Wrst StDev
0.0% 3 0.2 0.2 0.2 0.2 0.0
```

使用mtr命令检测网络稳定性结果中涉及的字段含义如下。

- Host: IP地址和域名, 按N键可以切换IP地址和域名
- Loss%: 检测数据包的丢包率
- Snt: 设置每秒发送数据包的数量, 默认值是10
- Last: 最近一次请求的时延
- Avg: 平均时延
- Best: 时延最短值
- Wrst: 时延最长值
- StDev: 时延标准偏差



2. 查看本地主机的信息

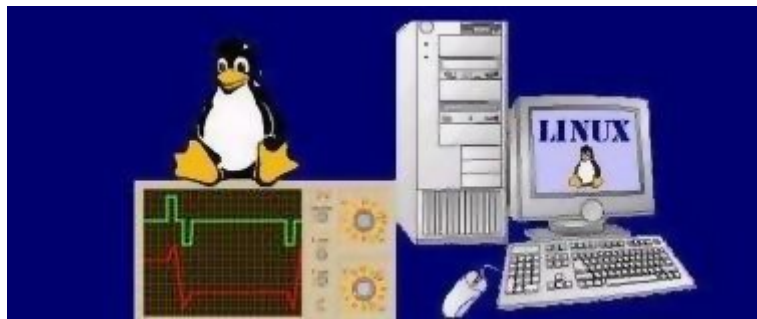
- mtr (My Traceroute) 是一种网络故障排查工具，可同时显示出 traceroute 和 ping 工具的检测结果，从而更加全面地分析网络故障。
 - mtr工具利用 ICMP 协议进行网络故障排查，其操作过程主要实现以下几个方面。
 - 发送 ICMP 数据包
 - mtr工具向目标主机发送一系列的 ICMP 数据包，这些数据包会在网络中跳跃，每经过一个路由节点就会返回一个响应信息。
 - 统计网络数据包的传输信息
 - mtr工具会记录数据包的丢失情况以及数据包的响应时间，从而可以分析网络延迟的情况。如果某个路由节点的响应时间过长，就说明这个路由节点可能存在故障。
 - 显示路由路径
 - mtr工具会显示数据包经过的路由路径，能帮助网络运维人员快速定位故障节点的位置。



3. 监控本地主机的性能

3.1 使用sysstat工具监控性能

- ❑ The sysstat utilities are a collection of performance monitoring tools for Linux.
- ❑ Sysstat is Open Source / Free Software, and is freely available under the GNU General Public License, version 2.



3. 监控本地主机的性能

3.1 使用sysstat工具监控性能

□ sysstat监控的指标

- Input / Output and transfer rate statistics (global, per device, per partition, per network filesystem and per Linux task / PID).
- CPU statistics (global, per CPU, per NUMA nodes and per Linux task / PID), including support for virtualization architectures.
- Memory, hugepages and swap space utilization statistics.
- Virtual memory, paging and fault statistics.
- Per-task (per-PID) memory and page fault statistics.
- Global CPU and page fault statistics for tasks and all their children.
- Process creation activity.
- Interrupt statistics (global, per CPU and per interrupt, including potential APIC interrupt sources, hardware and software interrupts).
- Extensive network statistics: network interface activity (number of packets and kB received and transmitted per second, etc.) including failures from network devices; network traffic statistics for IP, TCP, ICMP and UDP protocols based on SNMPv2 standards; support for IPv6-related protocols.
- Fibre Channel traffic statistics.
- Software-based network processing (softnet) statistics.
- NFS server and client activity.
- Sockets statistics.
- Run queue and system load statistics.
- Kernel internal tables utilization statistics.
- System and per Linux task switching activity.
- Swapping statistics.
- TTY devices activity.
- Power management statistics
- USB devices plugged into the system.
- Filesystems utilization (inodes and blocks).
- Tape drives statistics.
- Pressure-Stall Information statistics.

四组监控工具

sar / sadc / sadf
iostat / tapestat / cifsioat
mpstat
pidstat





Welcome to the
SYSSTAT
Utilities Home Page!



[News](#) | [Features](#) | [DTD/XSD](#) | [GitHub](#) | [Download](#) | [FAQ](#) | [Contact](#)

This page has news, information, documentation and links software for the **sysstat** utilities that I created for Linux. The sysstat utilities are a collection of **performance monitoring tools** for Linux. These include **sar**, **sadf**, **mpstat**, **iostat**, **tapestat**, **pidstat**, **cifsioostat** and **sa tools**. Go to the [Features](#) page to display a list of sysstat's features and learn some more about them. Sysstat is [Open Source](#) / [Free Software](#), and is freely available under the [GNU General Public License, version 2](#).

News

03 Jul 2024 - [sysstat 12.7.6 released](#)

This version brings improvements mainly to sadf's SVG output format. In particular the long-standing problem with **oversized SVG canvas height** has been fixed here. Also note that sar's **retrans/s** field used in the A_NET_ETCP report has been renamed to **retrseg/s** to avoid confusion with another field with the same name used in the A_NET_NFS report.

17 Dec 2023 - [sysstat 12.7.5 released](#)

The sar command here includes **two new metrics** (pgprom/s et pgdem/s) so that we can now track the number of pages promoted or demoted by the system per second (that is to say, the number of pages migrated from slow to fast, or from fast to slow, memory types). Another metric (%vmeff) has also been removed from sar, as it was more a kernel metric than a system one and it was wrongly calculated with recent kernels. You will also find various small fixes and improvements in this version.

18 Jun 2023 - [sysstat 12.7.4 released](#)

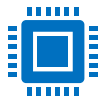
Previous 12.7.3 version (released two days ago) was a Brown Paper Bag release! Current one only fixes a problem that **prevented sysstat from being installed**. Sorry for that.

16 Jun 2023 - [sysstat 12.7.3 released](#)

This version adds **extended reports** support to sar. Extended reports, available using option -x with sar, display minimum and maximum values in addition to average ones at the end of each report. This version also includes a lot of fixes and small other improvements.

3. 监控本地主机的性能

3.1 使用sysstat工具监控性能



mpstat [选项] [参数]

功能:

- 监控系统 CPU。

参数/命令:

- 时间间隔:
 - 指定监控报告执行输出的时间间隔 (s)
- 次数:
 - 显示系统 CPU 检测的执行次数

主要选项:

- -P:
 - 指定 CPU 编号[0~n-1, 物理 CPU 编号从 0 开始],
 - 或者输入 ALL 表示监控所有 CPU

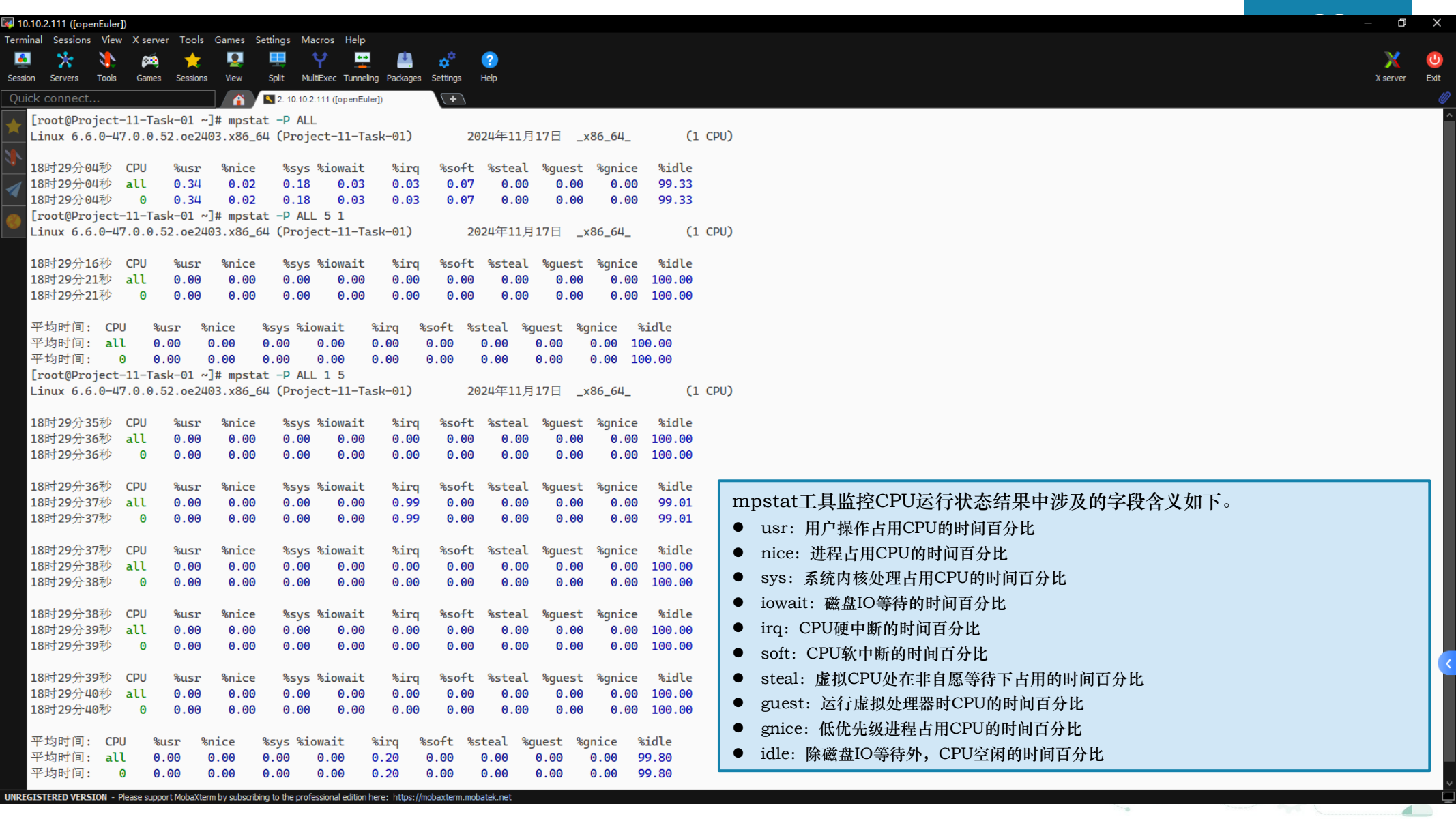


```
10.10.2.111 (openEuler)
Terminal Sessions View X server Tools Games Settings Macros Help
Session Servers Tools Games Sessions View Split MultiExec Tunneling Packages Settings Help
Quick connect... 2. 10.10.2.111 (openEuler)
[root@Project-11-Task-01 ~]# yum install -y sysstat
Last metadata expiration check: 0:22:20 ago on 2024年11月17日 星期日 18时05分51秒.
Dependencies resolved.
=====
Package                                Architecture      Version           Repository        Size
=====
Installing:
sysstat                                x86_64            12.7.5-2.oe2403  OS                456 k
Installing dependencies:
lm_sensors                              x86_64            3.6.0-8.oe2403   OS                139 k
=====
Transaction Summary
=====
Install 2 Packages

Total download size: 594 k
Installed size: 2.2 M
Downloading Packages:
(1/2): lm_sensors-3.6.0-8.oe2403.x86_64.rpm      724 kB/s | 139 kB    00:00
(2/2): sysstat-12.7.5-2.oe2403.x86_64.rpm      1.8 MB/s | 456 kB   00:00
=====
Total
Running transaction check
Transaction check succeeded.
Running transaction test
Transaction test succeeded.
Running transaction
  Preparing      :                                     1/1
  Installing     : lm_sensors-3.6.0-8.oe2403.x86_64  1/2
  Running scriptlet: lm_sensors-3.6.0-8.oe2403.x86_64 1/2
Created symlink /etc/systemd/system/multi-user.target.wants/lm_sensors.service → /usr/lib/systemd/system/lm_sensors.service.

  Running scriptlet: sysstat-12.7.5-2.oe2403.x86_64 2/2
  Installing     : sysstat-12.7.5-2.oe2403.x86_64 2/2
  Running scriptlet: sysstat-12.7.5-2.oe2403.x86_64 2/2
Created symlink /etc/systemd/system/multi-user.target.wants/sysstat.service → /usr/lib/systemd/system/sysstat.service.
Created symlink /etc/systemd/system/sysstat.service.wants/sysstat-collect.timer → /usr/lib/systemd/system/sysstat-collect.timer.
Created symlink /etc/systemd/system/sysstat.service.wants/sysstat-summary.timer → /usr/lib/systemd/system/sysstat-summary.timer.
Created symlink /etc/systemd/system/sysstat.service.wants/sysstat-rotate.timer → /usr/lib/systemd/system/sysstat-rotate.timer.

  Verifying      : lm_sensors-3.6.0-8.oe2403.x86_64 1/2
  Verifying      : sysstat-12.7.5-2.oe2403.x86_64 2/2
=====
```



```
[root@Project-11-Task-01 ~]# mpstat -P ALL
Linux 6.6.0-47.0.0.52.oe2403.x86_64 (Project-11-Task-01)      2024年11月17日  _x86_64_      (1 CPU)

18时29分04秒 CPU    %usr  %nice  %sys  %iowait  %irq  %soft  %steal  %guest  %gnice  %idle
18时29分04秒 all    0.34  0.02  0.18  0.03  0.03  0.07  0.00  0.00  0.00  99.33
18时29分04秒 0     0.34  0.02  0.18  0.03  0.03  0.07  0.00  0.00  0.00  99.33

[root@Project-11-Task-01 ~]# mpstat -P ALL 5 1
Linux 6.6.0-47.0.0.52.oe2403.x86_64 (Project-11-Task-01)      2024年11月17日  _x86_64_      (1 CPU)

18时29分16秒 CPU    %usr  %nice  %sys  %iowait  %irq  %soft  %steal  %guest  %gnice  %idle
18时29分21秒 all    0.00  0.00  0.00  0.00  0.00  0.00  0.00  0.00  0.00  100.00
18时29分21秒 0     0.00  0.00  0.00  0.00  0.00  0.00  0.00  0.00  0.00  100.00

平均时间: CPU    %usr  %nice  %sys  %iowait  %irq  %soft  %steal  %guest  %gnice  %idle
平均时间: all    0.00  0.00  0.00  0.00  0.00  0.00  0.00  0.00  0.00  100.00
平均时间: 0     0.00  0.00  0.00  0.00  0.00  0.00  0.00  0.00  0.00  100.00

[root@Project-11-Task-01 ~]# mpstat -P ALL 1 5
Linux 6.6.0-47.0.0.52.oe2403.x86_64 (Project-11-Task-01)      2024年11月17日  _x86_64_      (1 CPU)

18时29分35秒 CPU    %usr  %nice  %sys  %iowait  %irq  %soft  %steal  %guest  %gnice  %idle
18时29分36秒 all    0.00  0.00  0.00  0.00  0.00  0.00  0.00  0.00  0.00  100.00
18时29分36秒 0     0.00  0.00  0.00  0.00  0.00  0.00  0.00  0.00  0.00  100.00

18时29分36秒 CPU    %usr  %nice  %sys  %iowait  %irq  %soft  %steal  %guest  %gnice  %idle
18时29分37秒 all    0.00  0.00  0.00  0.00  0.99  0.00  0.00  0.00  0.00  99.01
18时29分37秒 0     0.00  0.00  0.00  0.00  0.99  0.00  0.00  0.00  0.00  99.01

18时29分37秒 CPU    %usr  %nice  %sys  %iowait  %irq  %soft  %steal  %guest  %gnice  %idle
18时29分38秒 all    0.00  0.00  0.00  0.00  0.00  0.00  0.00  0.00  0.00  100.00
18时29分38秒 0     0.00  0.00  0.00  0.00  0.00  0.00  0.00  0.00  0.00  100.00

18时29分38秒 CPU    %usr  %nice  %sys  %iowait  %irq  %soft  %steal  %guest  %gnice  %idle
18时29分39秒 all    0.00  0.00  0.00  0.00  0.00  0.00  0.00  0.00  0.00  100.00
18时29分39秒 0     0.00  0.00  0.00  0.00  0.00  0.00  0.00  0.00  0.00  100.00

18时29分39秒 CPU    %usr  %nice  %sys  %iowait  %irq  %soft  %steal  %guest  %gnice  %idle
18时29分40秒 all    0.00  0.00  0.00  0.00  0.00  0.00  0.00  0.00  0.00  100.00
18时29分40秒 0     0.00  0.00  0.00  0.00  0.00  0.00  0.00  0.00  0.00  100.00

平均时间: CPU    %usr  %nice  %sys  %iowait  %irq  %soft  %steal  %guest  %gnice  %idle
平均时间: all    0.00  0.00  0.00  0.00  0.20  0.00  0.00  0.00  0.00  99.80
平均时间: 0     0.00  0.00  0.00  0.00  0.20  0.00  0.00  0.00  0.00  99.80
```

mpstat工具监控CPU运行状态结果中涉及的字段含义如下。

- usr: 用户操作占用CPU的时间百分比
- nice: 进程占用CPU的时间百分比
- sys: 系统内核处理占用CPU的时间百分比
- iowait: 磁盘IO等待的时间百分比
- irq: CPU硬中断的时间百分比
- soft: CPU软中断的时间百分比
- steal: 虚拟CPU处在非自愿等待下占用的时间百分比
- guest: 运行虚拟处理器时CPU的时间百分比
- gnice: 低优先级进程占用CPU的时间百分比
- idle: 除磁盘IO等待外, CPU空闲的时间百分比

3. 监控本地主机的性能

3.1 使用sysstat工具监控性能



vmstat [选项] [参数]

功能:

- 统计系统整体的存储情况，包括内核进程、内存使用、虚拟内存、磁盘 IO 和 CPU 状态等信息。

参数/命令:

- 时间间隔:
 - 状态信息刷新的时间间隔 (单位: s)
- 次数:
 - 显示报告的次数

主要选项:

- -a: 显示活动和非活动内存
- -f: 显示启动后创建的进程总数
- -m: 显示 slab 信息 (内存分配机制)
- -n: 只在开始时显示一次各字段头信息
- -s: 以表格方式显示事件计数器和内存状态
- -d: 显示磁盘相关统计信息
- -p: 显示指定磁盘分区统计信息
- -S: 使用指定单位显示, 可使用 k、K、m、M



```
[root@Project-11-Task-01 ~]# vmstat
procs -----memory----- --swap-- ----io---- -system-- -----cpu-----
 r b swpd free buff cache si so bi bo in cs us sy id wa st gu
 1 0 268 266032 30464 337748 0 0 751 336 179 3 7 3 89 0 0 0
```

```
[root@Project-11-Task-01 ~]#
[root@Project-11-Task-01 ~]# vmstat -S M -s
 930 M total memory
 457 M used memory
 191 M active memory
 296 M inactive memory
 259 M free memory
 29 M buffer memory
 329 M swap cache
 2047 M total swap
 0 M used swap
 2047 M free swap
 2883 non-nice user cpu ticks
 14 nice user cpu ticks
 1184 system cpu ticks
 35451 idle cpu ticks
 112 IO-wait cpu ticks
 71 IRQ cpu ticks
 55 softirq cpu ticks
 0 stolen cpu ticks
 0 non-nice guest cpu ticks
 0 nice guest cpu ticks
 292618 K paged in
 131180 K paged out
 0 pages swapped in
 9 pages swapped out
 70228 interrupts
 149952 CPU context switches
 1731839452 boot time
 5422 forks
```

```
[root@Project-11-Task-01 ~]# vmstat 1 5
procs -----memory----- --swap-- ----io---- -system-- -----cpu-----
 r b swpd free buff cache si so bi bo in cs us sy id wa st gu
 1 0 268 261936 30480 337804 0 0 718 322 172 3 7 3 89 0 0 0
 0 0 268 261936 30480 337804 0 0 0 0 39 67 0 0 100 0 0 0
 0 0 268 261936 30480 337804 0 0 0 20 29 35 0 0 100 0 0 0
 0 0 268 261936 30480 337804 0 0 0 0 32 39 0 0 100 0 0 0
 0 0 268 261936 30480 337804 0 0 0 0 30 41 0 1 99 0 0 0
```

vmstat工具在操作过程中可查看系统的procs（进程）、memory（内存）、swap（交换分区）、io（磁盘IO）、system（系统中断）以及cpu的运行性能。

在监控系统存储的使用结果中，procs类型涉及的字段含义如下。

- r: 运行队列中进程的数量
- b: 等待IO的进程数量

在监控系统存储的使用结果中，memory类型涉及的字段含义如下。

- swpd: 虚拟内存使用量
- free: 空闲物理内存量
- buff: 用于缓冲的内存量
- cache: 用于缓存的内存量

在监控系统存储的使用结果中，swap类型涉及的字段含义如下。

- si: 每秒从交换分区写入内存数据量的大小
- so: 每秒写入交换分区数据量的大小

在监控系统存储的使用结果中，io类型涉及的字段含义如下。

- bi: 每秒读取的磁盘块数
- bo: 每秒写入的磁盘块数

在监控系统存储的使用结果中，system类型涉及的字段含义如下。

- in: 每秒系统中断数
- cs: 每秒上下文切换数

在监控系统存储的使用结果中，cpu类型涉及的字段含义如下。

- us: 用户进程执行时间百分比
- sy: 内核系统进程执行时间百分比
- wa: IO等待时间百分比
- id: CPU空闲时间百分比

3. 监控本地主机的性能

3.1 使用sysstat工具监控性能



pidstat [选项] [参数]

功能:

- 监控全部或单独指定某个进程，查看其资源占用情况，掌握系统进程的运行性能。

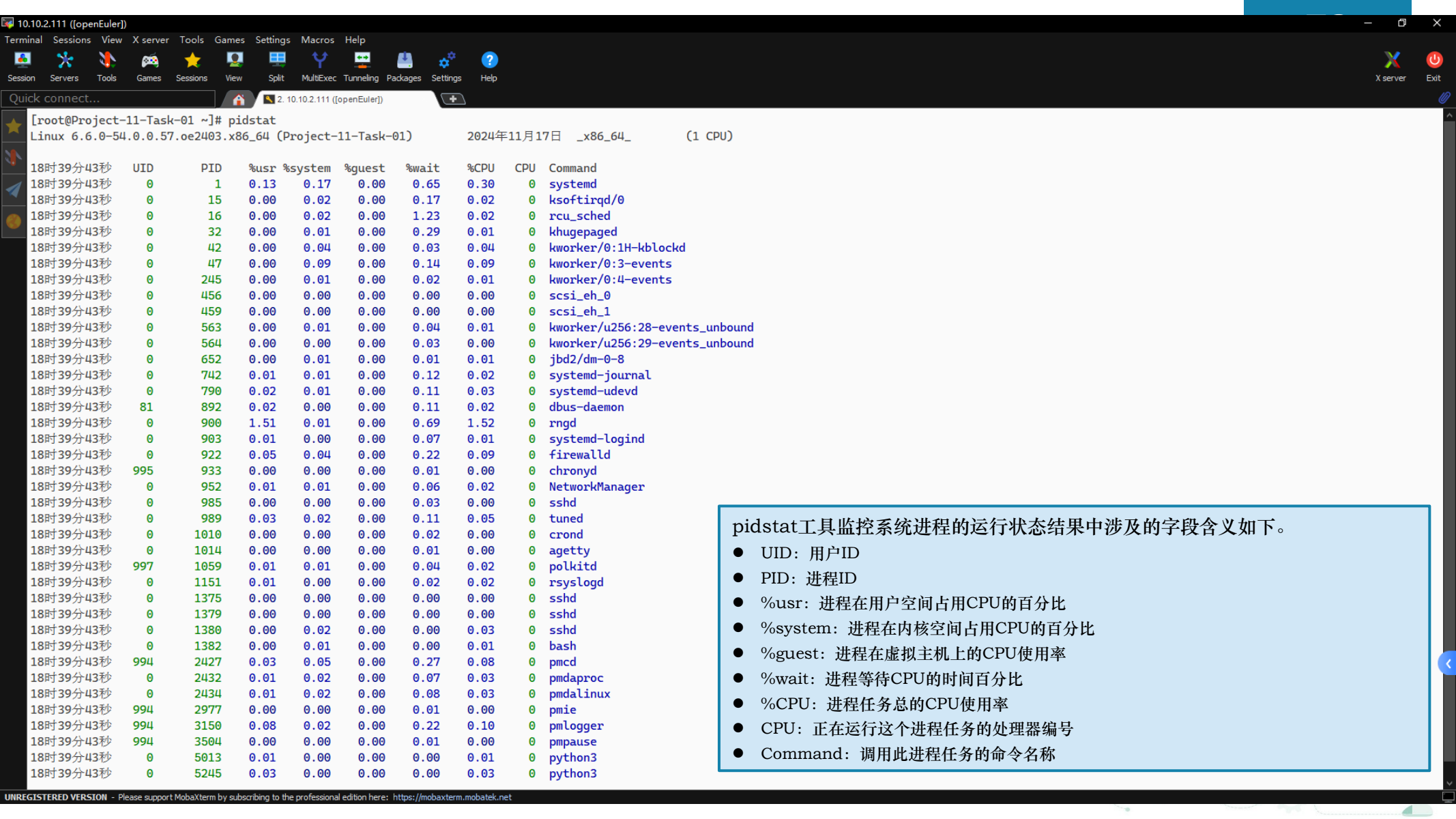
参数/命令:

- 时间间隔:
 - 指定监控报告执行输出的时间间隔 (s)
- 次数:
 - 显示系统进程检测的执行次数

主要选项:

- -u: 默认的参数，显示各个进程的 CPU 使用统计
- -r: 显示各个进程的内存使用统计
- -d: 显示各个进程的 IO 使用情况
- -p: 指定进程号
- -w: 显示每个进程的上下文切换情况
- -t: 显示选择任务的线程的统计信息外的额外信息





```
[root@Project-11-Task-01 ~]# pidstat
Linux 6.6.0-54.0.0.57.oe2403.x86_64 (Project-11-Task-01)      2024年11月17日  _x86_64_      (1 CPU)
```

	UID	PID	%usr	%system	%guest	%wait	%CPU	CPU	Command
18时39分43秒	0	1	0.13	0.17	0.00	0.65	0.30	0	systemd
18时39分43秒	0	15	0.00	0.02	0.00	0.17	0.02	0	ksoftirqd/0
18时39分43秒	0	16	0.00	0.02	0.00	1.23	0.02	0	rcu_sched
18时39分43秒	0	32	0.00	0.01	0.00	0.29	0.01	0	khugepaged
18时39分43秒	0	42	0.00	0.04	0.00	0.03	0.04	0	kworker/0:1H-kblockd
18时39分43秒	0	47	0.00	0.09	0.00	0.14	0.09	0	kworker/0:3-events
18时39分43秒	0	245	0.00	0.01	0.00	0.02	0.01	0	kworker/0:4-events
18时39分43秒	0	456	0.00	0.00	0.00	0.00	0.00	0	scsi_eh_0
18时39分43秒	0	459	0.00	0.00	0.00	0.00	0.00	0	scsi_eh_1
18时39分43秒	0	563	0.00	0.01	0.00	0.04	0.01	0	kworker/u256:28-events_unbound
18时39分43秒	0	564	0.00	0.00	0.00	0.03	0.00	0	kworker/u256:29-events_unbound
18时39分43秒	0	652	0.00	0.01	0.00	0.01	0.01	0	jbd2/dm-0-8
18时39分43秒	0	742	0.01	0.01	0.00	0.12	0.02	0	systemd-journal
18时39分43秒	0	790	0.02	0.01	0.00	0.11	0.03	0	systemd-udev
18时39分43秒	81	892	0.02	0.00	0.00	0.11	0.02	0	dbus-daemon
18时39分43秒	0	900	1.51	0.01	0.00	0.69	1.52	0	rngd
18时39分43秒	0	903	0.01	0.00	0.00	0.07	0.01	0	systemd-logind
18时39分43秒	0	922	0.05	0.04	0.00	0.22	0.09	0	firewalld
18时39分43秒	995	933	0.00	0.00	0.00	0.01	0.00	0	chronyd
18时39分43秒	0	952	0.01	0.01	0.00	0.06	0.02	0	NetworkManager
18时39分43秒	0	985	0.00	0.00	0.00	0.03	0.00	0	sshd
18时39分43秒	0	989	0.03	0.02	0.00	0.11	0.05	0	tuned
18时39分43秒	0	1010	0.00	0.00	0.00	0.02	0.00	0	crond
18时39分43秒	0	1014	0.00	0.00	0.00	0.01	0.00	0	agetty
18时39分43秒	997	1059	0.01	0.01	0.00	0.04	0.02	0	polkitd
18时39分43秒	0	1151	0.01	0.00	0.00	0.02	0.02	0	rsyslogd
18时39分43秒	0	1375	0.00	0.00	0.00	0.00	0.00	0	sshd
18时39分43秒	0	1379	0.00	0.00	0.00	0.00	0.00	0	sshd
18时39分43秒	0	1380	0.00	0.02	0.00	0.00	0.03	0	sshd
18时39分43秒	0	1382	0.00	0.01	0.00	0.00	0.01	0	bash
18时39分43秒	994	2427	0.03	0.05	0.00	0.27	0.08	0	pmcd
18时39分43秒	0	2432	0.01	0.02	0.00	0.07	0.03	0	pmdaproc
18时39分43秒	0	2434	0.01	0.02	0.00	0.08	0.03	0	pmdalinux
18时39分43秒	994	2977	0.00	0.00	0.00	0.01	0.00	0	pmie
18时39分43秒	994	3150	0.08	0.02	0.00	0.22	0.10	0	pmlogger
18时39分43秒	994	3504	0.00	0.00	0.00	0.01	0.00	0	pmpause
18时39分43秒	0	5013	0.01	0.00	0.00	0.00	0.01	0	python3
18时39分43秒	0	5245	0.03	0.00	0.00	0.00	0.03	0	python3

pidstat工具监控系统进程的运行状态结果中涉及的字段含义如下。

- UID: 用户ID
- PID: 进程ID
- %usr: 进程在用户空间占用CPU的百分比
- %system: 进程在内核空间占用CPU的百分比
- %guest: 进程在虚拟主机上的CPU使用率
- %wait: 进程等待CPU的时间百分比
- %CPU: 进程任务总的CPU使用率
- CPU: 正在运行这个进程任务的处理器编号
- Command: 调用此进程任务的命令名称

3. 监控本地主机的性能

3.1 使用sysstat工具监控性能



iostat [选项] [参数]

功能:

- 监控主机磁盘 IO 的运行情况，查看存储设备的性能。

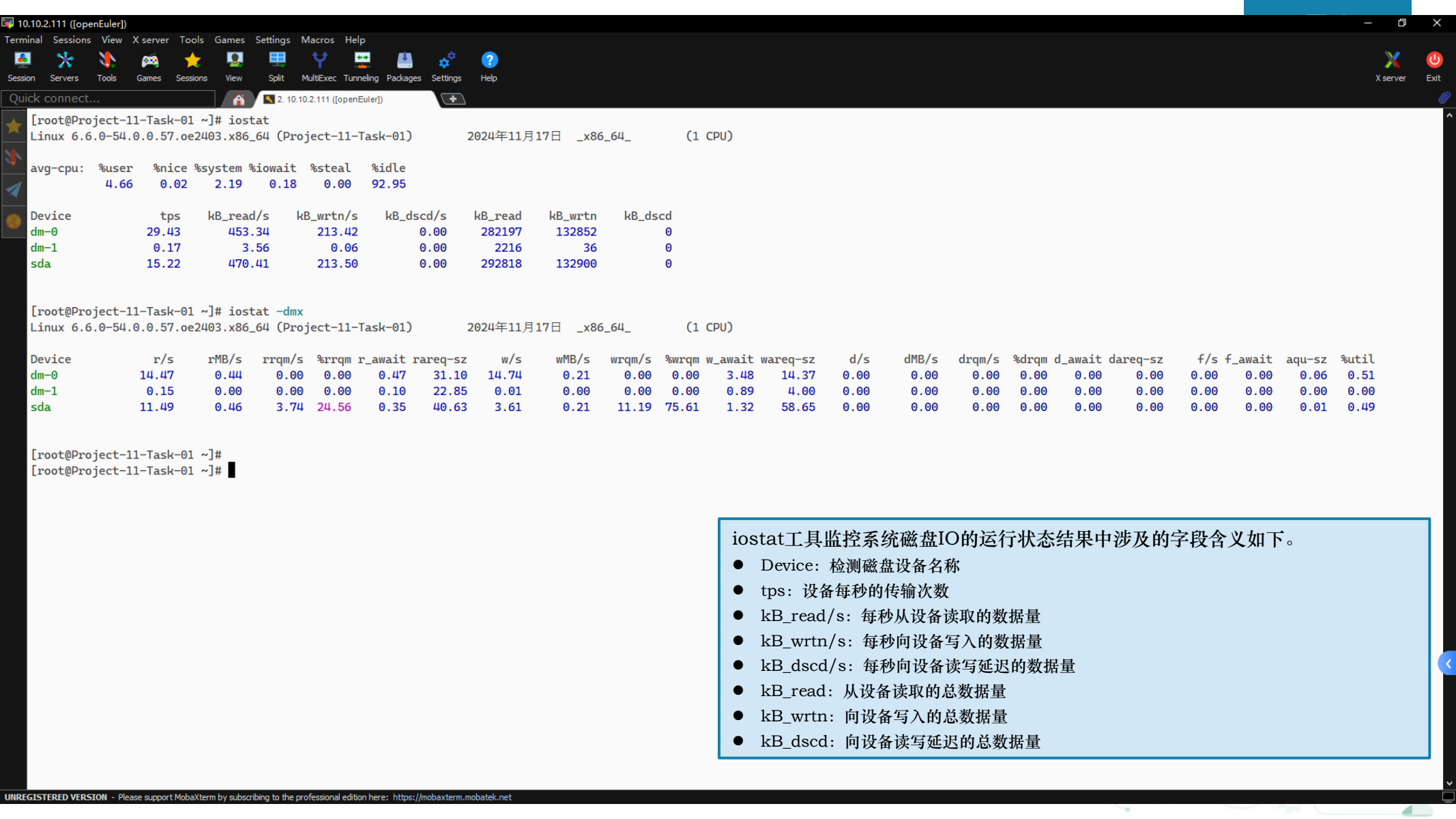
参数/命令:

- 时间间隔:
 - 每次报告产生的间隔时间 (s)
- 次数:
 - 显示报告的次数

主要选项:

- -c: 仅显示 CPU 使用情况
- -d: 仅显示磁盘设备 IO 情况
- -k: 显示状态以千字节每秒为单位，而不使用块每秒
- -m: 显示状态以兆字节每秒为单位
- -p: 仅显示块设备和所有被使用的其他分区状态
- -t: 显示每个报告产生的时间
- -x: 显示扩展状态信息





```
[root@Project-11-Task-01 ~]# iostat
Linux 6.6.0-54.0.0.57.oe2403.x86_64 (Project-11-Task-01)      2024年11月17日  _x86_64_      (1 CPU)
```

```
avg-cpu:  %user   %nice %system %iowait  %steal   %idle
            4.66    0.02   2.19   0.18    0.00   92.95
```

Device	tps	kB_read/s	kB_wrtn/s	kB_dscd/s	kB_read	kB_wrtn	kB_dscd
dm-0	29.43	453.34	213.42	0.00	282197	132852	0
dm-1	0.17	3.56	0.06	0.00	2216	36	0
sda	15.22	470.41	213.50	0.00	292818	132900	0

```
[root@Project-11-Task-01 ~]# iostat -dmx
Linux 6.6.0-54.0.0.57.oe2403.x86_64 (Project-11-Task-01)      2024年11月17日  _x86_64_      (1 CPU)
```

Device	r/s	rMB/s	rrqm/s	%rrqm	r_await	rareq-sz	w/s	wMB/s	wrqm/s	%wrqm	w_await	wareq-sz	d/s	dMB/s	drqm/s	%drqm	d_await	dareq-sz	f/s	f_await	aqu-sz	%util
dm-0	14.47	0.44	0.00	0.00	0.47	31.10	14.74	0.21	0.00	0.00	3.48	14.37	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.06	0.51
dm-1	0.15	0.00	0.00	0.00	0.10	22.85	0.01	0.00	0.00	0.00	0.89	4.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
sda	11.49	0.46	3.74	24.56	0.35	40.63	3.61	0.21	11.19	75.61	1.32	58.65	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.01	0.49

```
[root@Project-11-Task-01 ~]#
[root@Project-11-Task-01 ~]# █
```

iostat工具监控系统磁盘IO的运行状态结果中涉及的字段含义如下。

- Device: 检测磁盘设备名称
- tps: 设备每秒的传输次数
- kB_read/s: 每秒从设备读取的数据量
- kB_wrtn/s: 每秒向设备写入的数据量
- kB_dscd/s: 每秒向设备读写延迟的数据量
- kB_read: 从设备读取的总数据量
- kB_wrtn: 向设备写入的总数据量
- kB_dscd: 向设备读写延迟的总数据量

3. 监控本地主机的性能

3.2 访问PROC获取指标数据

- proc是伪文件系统（即虚拟文件系统）
 - 只存在内存中，是存储当前内核运行状态的一系列特殊文件，用户可通过该类型文件查看主机以及当前正在运行进程的信息，甚至可以通过更改其中某些文件来改变内核的运行状态。
 - proc目录下的文件也常被称为虚拟文件，通常文件的时间及日期属性为当前系统的时间和日期，虚拟文件是随时刷新的。
 - proc目录下的文件通常是只读属性。
 - 与proc下其他文件的只读属性不同，管理员可对/proc/sys 子目录中的许多文件内容进行修改，通过此更改可以调整内核的运行特性。



表 12-6-1 /proc 下常见的目录

目录	描述
/proc/apm	高级电源管理（APM）版本信息及电池相关状态信息，通常由 apm 命令使用
/proc/buddyinfo	用于诊断内存碎片问题的相关信息
/proc/cmdline	在启动时传递至内核的相关参数信息，这些信息通常由 lilo（Linux 加载程序）或 grub（Linux 引导管理程序）等工具进行传递
/proc/cpuinfo	处理器的相关信息文件
/proc/crypto	系统上已安装内核使用的密码算法及每个算法的详细信息列表
/proc/devices	系统已经加载的所有块设备和字符设备的信息，包含主设备号和设备组（与主设备号对应的设备类型）名
/proc/diskstats	每块磁盘设备的 I/O 统计信息列表（内核 2.5.69 以后的版本支持此功能）
/proc/dma	每个正在使用且注册的 ISA DMA 通道信息列表
/proc/execdomains	内核当前支持的执行域信息列表
/proc/fb	帧缓冲设备列表文件，包含帧缓冲设备的设备号和相关驱动信息
/proc/filesystems	当前被内核支持的文件系统类型列表文件，被标示为 nodev 的文件系统表示不需要该块设备的支持；通常“mount”设备时，如果没有指定文件系统类型，将通过此文件来决定其所需文件系统的类型
/proc/interrupts	X86 或 X86_64 体系架构系统上每个 IRQ（Interrupt Request，中断请求）相关的中断信息列表
/proc/iomem	每个物理设备上的记忆体（RAM 或者 ROM）在系统内存中的映射信息
/proc/ioports	当前正在使用且已经被注册过的与物理设备进行通讯的输入-输出端口范围信息列表
/proc/kallsyms	模块管理工具，用来动态链接或绑定可装载模块的符号定义，由内核输出（内核 2.5.71 以后的版本支持此功能），通常这个文件中的信息量较大



表 12-6-1 /proc 下常见的目录

目录	描述
/proc/kcore	系统使用的物理内存以 ELF 核心文件（core file）格式存储，其文件大小为已使用物理内存加上 4KB；此文件用来检查内核数据结构的当前状态，通常由 GDB 调试工具使用，但不能使用文件查看命令打开此文件
/proc/kmsg	此文件用来保存由内核输出的信息，通常由/sbin/klogd 或/bin/dmmsg 等程序使用，不能使用文件查看命令打开此文件
/proc/loadavg	保存关于 CPU 和磁盘 I/O 的负载平均值，其前三列分别表示每 1 分钟、每 5 分钟及每 15 分钟的负载平均值，类似于 uptime 命令输出的相关信息；第四列是由斜线隔开的两个数值，前者表示当前正由内核调度的实体（进程和线程）的数目，后者表示系统当前存活的内核调度实体的数目；第五列表示此文件被查看前最近一个由内核创建的进程 PID
/proc/locks	保存当前由内核锁定的文件相关信息，包含内核内部的调试数据；每个锁定占据一行，且具有一个唯一的编号；输出信息中每行的第二列表示当前锁定使用的锁定类别，POSIX 表示目前较新类型的文件锁，有 lockf 系统调用产生，FLOCK 是传统的 UNIX 文件锁，由 flock 系统调用产生；第三列也通常由两种类型，ADVISORY 表示不允许其他用户锁定此文件，但允许读取，MDNDATOR Y 表示此文件锁定期间不允许其他用户以任何形式的访问
/proc/mdstat	保存 RAID 相关的多块磁盘的当前状态信息，在没有使用 RAID 机器上，其显示为<none>
/proc/meminfo	系统中关于当前内存的利用状况等的信息，常由 free 命令使用；可以使用文件查看命令直接读取，其内容显示为两列，前者为统计属性，后者为对应的值
/proc/mounts	在内核 2.4.29 版本以前，此文件的内容为系统当前挂载的所有文件系统，在 2.4.29 以后的内核中引进了每个进程使用独立挂载名称空间的方式，此文件则随之变成了指向/proc/self/mounts（每个进程自身挂载名称空间中的所有挂载点列表）文件的符号链接。

表 12-6-1 /proc 下常见的目录

目录	描述
/proc/partitions	块设备每个分区的主设备号（major）和次设备号（minor）等信息，同时包括每个分区所包含的块（block）数目
/proc/pci	内核初始化时发现的所有 PCI 设备及其配置信息列表，其配置信息多为某 PCI 设备相关 IRQ 信息，可读性不高，可以用“/sbin/lspci -vb”命令获得较易理解的相关信息。在内核 2.6 版本以后，此文件已为/proc/bus/pci 目录及其下的文件代替
/proc/slabinfo	在内核中频繁使用的对象（如 inode、dentry 等）都有相应的 cache，即 slab pool，而/proc/slabinfo 文件列出了这些对象相关 slab 信息
/proc/stat	实时追踪自系统上次启动以来的多种统计信息，其中具体每行含义如表 12-6-2 所示
/proc/swaps	当前系统上的交换分区及其空间利用信息，如果有多个交换分区的话，则会将每个交换分区的信息分别存储于/proc/swap 目录中的单独文件中，而其优先级数字越低，被使用到的可能性越大
/proc/uptime	系统上次启动以来的运行时间，其第一个数字表示系统运行时间，第二个数字表示系统空闲时间，单位是秒
/proc/version	当前系统运行的内核版本号
/proc/vmstat	当前系统虚拟内存的统计数据，可读性较好（内核 2.6 版本以后支持此文件）
/proc/zoneinfo	内存区域（zone）的详细信息列表



表 12-6-2 /proc/stat 信息内容

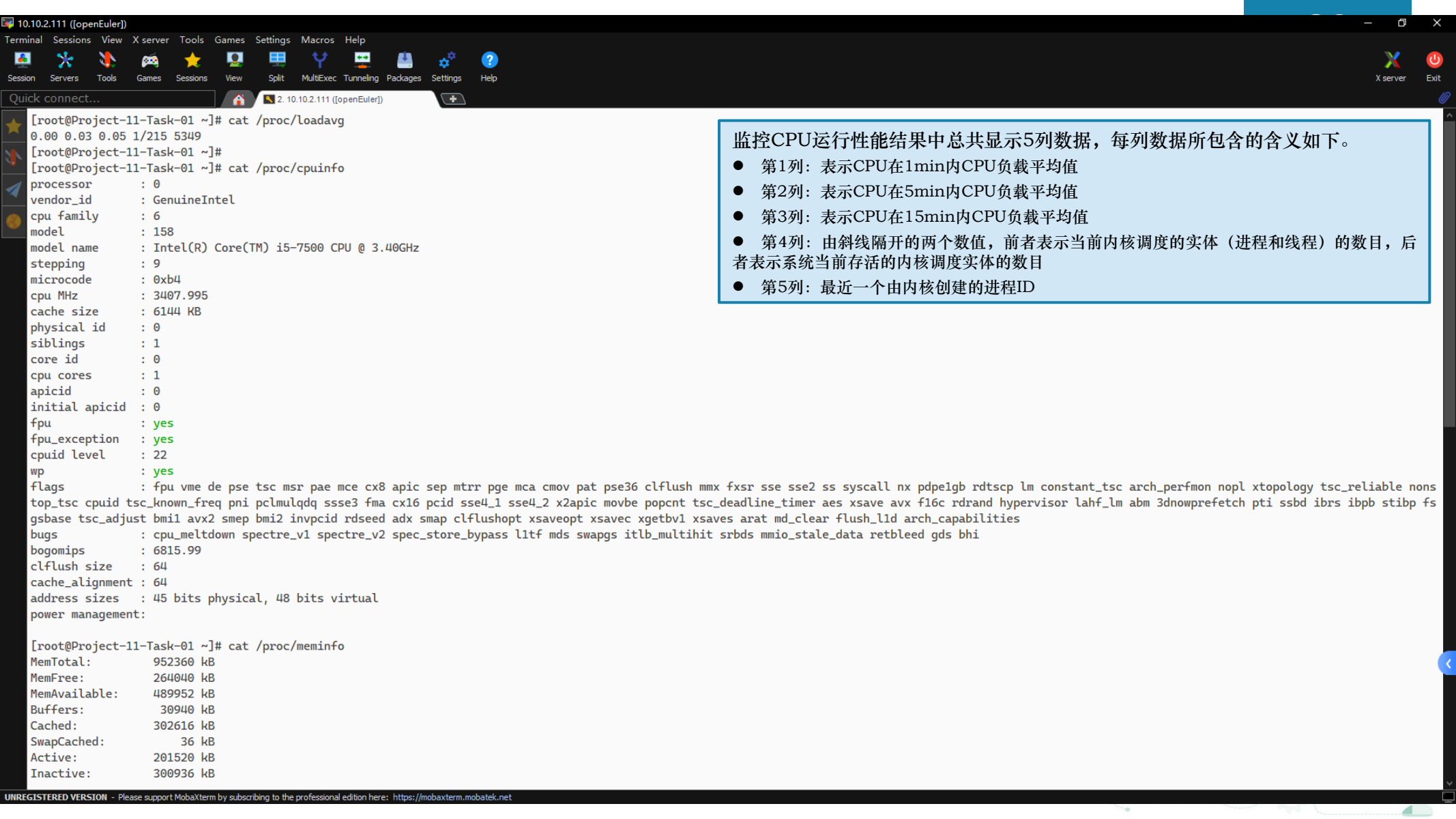
行名	描述
cpu	该行后的八个值分别表示以 1/100 (jiffies) 秒为单位的统计值 (包括系统运行于用户模式、低优先级用户模式、系统模式、空闲模式、I/O 等待模式的时间等)
intr	该行给出中断的信息, 第一个为自系统启动以来, 发生的所有的中断的次数; 然后每个数对应一个特定的中断自系统启动以来所发生的次数
ctxt	该行展示从系统启动以来 CPU 发生的上下文交换的次数
btime	该行展示从系统启动到现在为止的时间, 单位为秒
processes (total_forks)	该行展示从系统启动以来所创建的任务的个数目
procs_running	该行展示当前运行队列的任务数目
procs_blocked	该行展示当前被阻塞的任务数目

表 12-6-3 /proc/sys 系统目录内容

目录	描述
/proc/sys/abi	此目录主要记录应用程序二进制接口, 涉及了程序的多个方面, 如目标文件格式、数据类型、函数调用以及函数传递参数等信息
/proc/sys/crypto	此目录主要记录系统中已经安装的相关服务使用的信息加密处理配置
/proc/sys/debug	此目录主要记录系统运行中的调试信息, 此目录通常是一空目录
/proc/sys/dev	为系统上特殊设备提供参数信息文件的目录, 其不同设备的信息文件分别存储于不同的子目录中, 如大多数系统上都会具有的/proc/sys/dev/cdrom 和/proc/sys/dev/raid (如果内核编译时开启了支持 raid 的功能) 目录, 其内存储的通常是系统上 cdrom 和 raid 的相关参数信息文件
/proc/sys/fs	该目录包含一系列选项以及有关文件系统的各个方面信息, 包括配额、文件句柄、索引以及系统登录信息
/proc/sys/kernel	此目录文件可用于监视和调整 Linux 操作中的内核相关参数
/proc/sys/net	主要包括了许多网络相关的操作, 如 appletalk/、ethernet/、ipv4/、ipx/ 及 ipv6/等, 通过改变这些目录中文件, 能够在系统运行时调整相关网络参数
/proc/sys/vm	该目录下文件主要用来优化系统中的虚拟内存



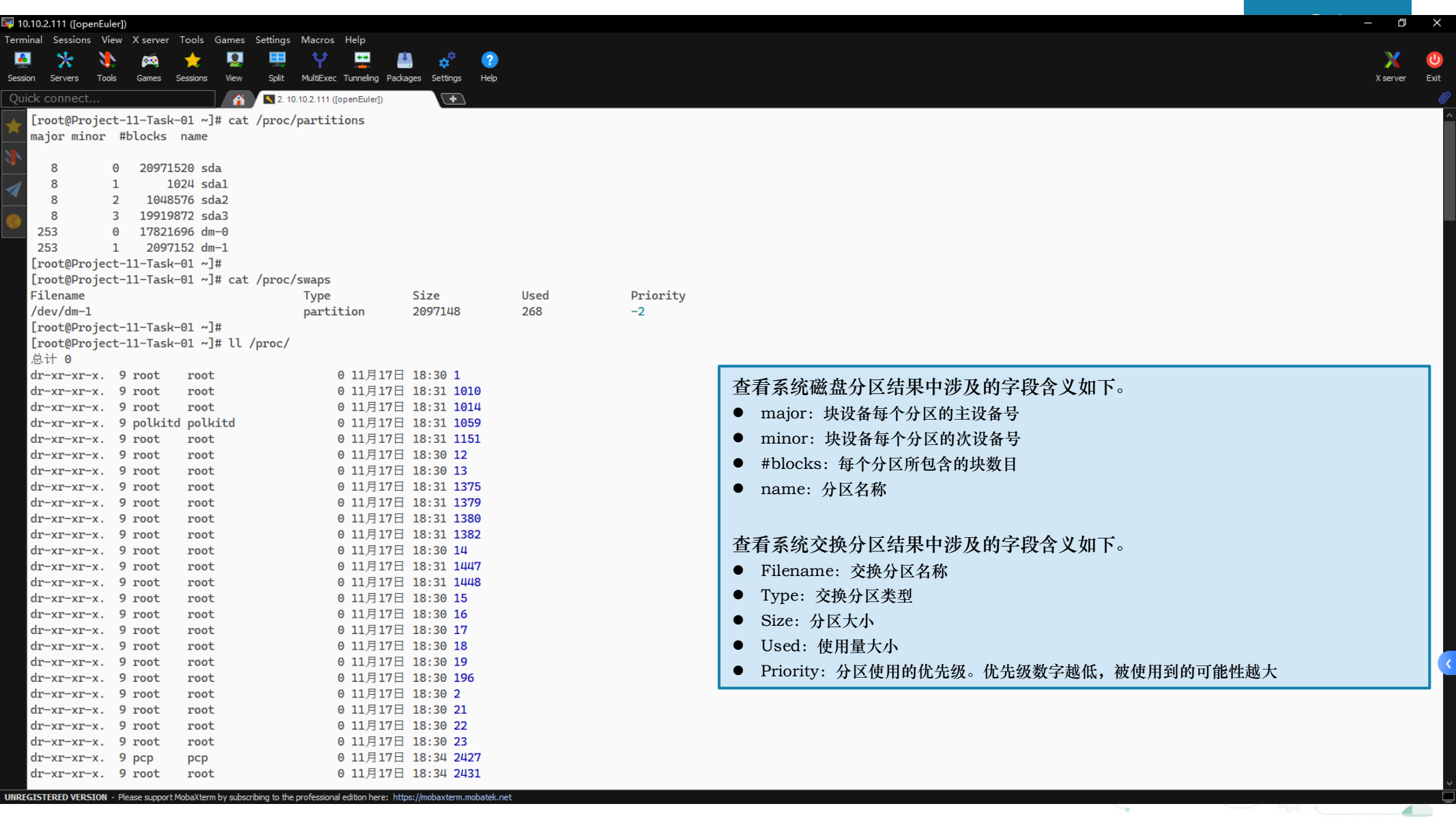
```
10.10.2.111 (openEuler)
Terminal Sessions View X server Tools Games Settings Macros Help
Session Servers Tools Games Sessions View Split MultiExec Tunneling Packages Settings Help
Quick connect... 2. 10.10.2.111 (openEuler)
[root@Project-11-Task-01 ~]# tree -L 1 -D --sort name -r /proc
[Nov 17 18:30] /proc
-- [Nov 17 18:34] zoneinfo
-- [Nov 17 18:34] vmstat
-- [Nov 17 18:42] vmallocinfo
-- [Nov 17 18:42] version
-- [Nov 17 18:34] uptime
-- [Nov 17 18:31] tty
-- [Nov 17 18:42] timer_list
-- [Nov 17 18:30] thread-self -> 5340/task/5340
-- [Nov 17 18:34] sysvipc
-- [Nov 17 18:42] sysrq-trigger
-- [Nov 17 18:30] sys
-- [Nov 17 18:30] swaps
-- [Nov 17 18:31] stat
-- [Nov 17 18:34] softirqs
-- [Nov 17 18:42] slabinfo
-- [Nov 17 18:30] self -> 5340
-- [Nov 17 18:34] scsi
-- [Nov 17 18:42] schedstat
-- [Nov 17 18:42] partitions
-- [Nov 17 18:42] pagetypeinfo
-- [Nov 17 18:30] net -> self/net
-- [Nov 17 18:42] mtrr
-- [Nov 17 18:42] mpt
-- [Nov 17 18:30] mounts -> self/mounts
-- [Nov 17 18:42] modules
-- [Nov 17 18:30] misc
-- [Nov 17 18:30] meminfo
-- [Nov 17 18:42] mdstat
-- [Nov 17 18:42] locks
-- [Nov 17 18:31] loadavg
-- [Nov 17 18:42] livepatch
-- [Nov 17 18:42] kpageflags
-- [Nov 17 18:42] kpagecount
-- [Nov 17 18:42] kpagecgroup
-- [Nov 17 18:42] kmsg
-- [Nov 17 18:42] key-users
-- [Nov 17 18:42] keys
-- [Nov 17 18:30] kcore
-- [Nov 17 18:42] kallsyms
-- [Nov 17 18:31] irq
```



```
[root@Project-11-Task-01 ~]# cat /proc/loadavg
0.00 0.03 0.05 1/215 5349
[root@Project-11-Task-01 ~]#
[root@Project-11-Task-01 ~]# cat /proc/cpuinfo
processor       : 0
vendor_id     : GenuineIntel
cpu family    : 6
model        : 158
model name    : Intel(R) Core(TM) i5-7500 CPU @ 3.40GHz
stepping     : 9
microcode    : 0xb4
cpu MHz      : 3407.995
cache size   : 6144 KB
physical id  : 0
siblings     : 1
core id      : 0
cpu cores    : 1
apicid       : 0
initial apicid : 0
fpu          : yes
fpu_exception : yes
cpuid level  : 22
wp           : yes
flags        : fpu vme de pse tsc msr pae mce cx8 apic sep mtrr pge mca cmov pat pse36 clflush mmx fxsr sse sse2 ss syscall nx pdpe1gb rdtscp lm constant_tsc arch_perfmon nopl xtopology tsc_reliable nons
top_tsc cpuid tsc_known_freq pni pclmulqdq ssse3 fma cx16 pcid sse4_1 sse4_2 x2apic movbe popcnt tsc_deadline_timer aes xsave avx f16c rdrand hypervisor lahf_lm abm 3dnowprefetch pti ssbd ibrs ibpb stibp fs
gsbase tsc_adjust bmi1 avx2 smep bmi2 invpcid rdseed adx smap clflushopt xsaveopt xsaves arat md_clear flush_l1d arch_capabilities
bugs         : cpu_meltdown spectre_v1 spectre_v2 spec_store_bypass l1tf mds swaps itlb_multihit srbds mmio_stale_data retbleed gds bhi
bogomips     : 6815.99
clflush size : 64
cache_alignm : 64
address sizes : 45 bits physical, 48 bits virtual
power management:

[root@Project-11-Task-01 ~]# cat /proc/meminfo
MemTotal:      952360 kB
MemFree:       264040 kB
MemAvailable:  489952 kB
Buffers:       30940 kB
Cached:        302616 kB
SwapCached:    36 kB
Active:        201520 kB
Inactive:      300936 kB
```

- 监控CPU运行性能结果中总共显示5列数据，每列数据所包含的含义如下。
- 第1列：表示CPU在1min内CPU负载平均值
 - 第2列：表示CPU在5min内CPU负载平均值
 - 第3列：表示CPU在15min内CPU负载平均值
 - 第4列：由斜线隔开的两个数值，前者表示当前内核调度的实体（进程和线程）的数目，后者表示系统当前存活的内核调度实体的数目
 - 第5列：最近一个由内核创建的进程ID



```
[root@Project-11-Task-01 ~]# cat /proc/partitions
major minor #blocks name
```

```
8      0   20971520 sda
8      1     1024 sda1
8      2   1048576 sda2
8      3   19919872 sda3
253    0   17821696 dm-0
253    1    2097152 dm-1
```

```
[root@Project-11-Task-01 ~]#
```

```
[root@Project-11-Task-01 ~]# cat /proc/swaps
```

```
Filename                Type      Size      Used      Priority
/dev/dm-1                partition 2097148    268       -2
```

```
[root@Project-11-Task-01 ~]#
```

```
[root@Project-11-Task-01 ~]# ll /proc/
```

```
总计 0
dr-xr-xr-x. 9 root  root      0 11月17日 18:30 1
dr-xr-xr-x. 9 root  root      0 11月17日 18:31 1010
dr-xr-xr-x. 9 root  root      0 11月17日 18:31 1014
dr-xr-xr-x. 9 polkitd polkitd    0 11月17日 18:31 1059
dr-xr-xr-x. 9 root  root      0 11月17日 18:31 1151
dr-xr-xr-x. 9 root  root      0 11月17日 18:30 12
dr-xr-xr-x. 9 root  root      0 11月17日 18:30 13
dr-xr-xr-x. 9 root  root      0 11月17日 18:31 1375
dr-xr-xr-x. 9 root  root      0 11月17日 18:31 1379
dr-xr-xr-x. 9 root  root      0 11月17日 18:31 1380
dr-xr-xr-x. 9 root  root      0 11月17日 18:31 1382
dr-xr-xr-x. 9 root  root      0 11月17日 18:30 14
dr-xr-xr-x. 9 root  root      0 11月17日 18:31 1447
dr-xr-xr-x. 9 root  root      0 11月17日 18:31 1448
dr-xr-xr-x. 9 root  root      0 11月17日 18:30 15
dr-xr-xr-x. 9 root  root      0 11月17日 18:30 16
dr-xr-xr-x. 9 root  root      0 11月17日 18:30 17
dr-xr-xr-x. 9 root  root      0 11月17日 18:30 18
dr-xr-xr-x. 9 root  root      0 11月17日 18:30 19
dr-xr-xr-x. 9 root  root      0 11月17日 18:30 196
dr-xr-xr-x. 9 root  root      0 11月17日 18:30 2
dr-xr-xr-x. 9 root  root      0 11月17日 18:30 21
dr-xr-xr-x. 9 root  root      0 11月17日 18:30 22
dr-xr-xr-x. 9 root  root      0 11月17日 18:30 23
dr-xr-xr-x. 9 pcp   pcp      0 11月17日 18:34 2427
dr-xr-xr-x. 9 root  root      0 11月17日 18:34 2431
```

查看系统磁盘分区结果中涉及的字段含义如下。

- major: 块设备每个分区的主设备号
- minor: 块设备每个分区的次设备号
- #blocks: 每个分区所包含的块数目
- name: 分区名称

查看系统交换分区结果中涉及的字段含义如下。

- Filename: 交换分区名称
- Type: 交换分区类型
- Size: 分区大小
- Used: 使用量大小
- Priority: 分区使用的优先级。优先级数字越低, 被使用到的可能性越大

4. 实现可视化系统监控

- Linux-Dash是一个监控Linux服务器的Web化开源工具，可监控主机CPU、内存、负载、网络、磁盘、用户等系统的运行状况。
- Linux-Dash的主要特性。
 - 具有基于Web的美观、简单易读的Linux服务器监控面板。
 - 工具文件较小，通过GitHub下载不到1MB。
 - 支持添加新的监控模块与实时数据可视化更新。
 - 支持在PHP、Node.js、Python和Go环境下安装。
- Linux-Dash:
 - A **simple & low-overhead web dashboard** for linux systems。

基于Web的仪表盘

Linux Dash – 轻量级但功能强大的 Linux 系统监控工具: <https://www.linuxmi.com/linux-dash-system-monitoring-tool.html>



GitHub - tariqbuilds/linux-dash: A simple & low-overhead web dashboard for Linux systems

Product Solutions Resources Open Source Enterprise Pricing

tariqbuilds / linux-dash Public

Code Issues 31 Pull requests 13 Actions Projects Wiki Security Insights

master Go to file

app Merge pull request #462 from Artenek... 4 years ago

bin Merge branch 'v2.0' 7 years ago

src Merge branch 'v2.0' 7 years ago

.gitignore Merge branch 'v2.0' 7 years ago

LICENSE.md updated application to work with segre... 10 years ago

README.md Updated the README with the host do... 6 years ago

demo.js finished first-pass through all pages of ... 7 years ago

ecosystem.config.js added ecosystem file for PM2 4 years ago

gulpfile.js preserve routing when using http fallba... 7 years ago

index.html added first-pass of demo 7 years ago

package.json last minute cleanup of Readme.md 7 years ago

README MIT license

v2.0
A simple & low-overhead web dashboard for Linux systems

[Demo](#) | [Docs](#)

chat on gitter

Features

- **Small** ----- Under 400KB on disk (*with .git removed!*)
- **Simple** ----- A minimalist, beautiful dashboard
- **Easy** ----- Drop-in installation
- **Versatile** -- Choose your stack from Node.js, Go, Python, PHP

Contributors 90

Languages

- JavaScript 60.8%
- Shell 20.6%
- HTML 7.4%
- CSS 7.4%
- Python 2.1%
- Go 1.4%
- PHP 0.3%

Linux Dash: Simple, beautiful server monitoring dashboard

Linux Dash SYSTEM STATUS BASIC INFO NETWORK ACCOUNTS APPS

Resources: CPU, Memory, Disk, Network, Accounts, Apps

RAM Usage

Used: 200 MB (39%)
Available: 312 MB of 512 MB

CPU Avg Load

1_min_avg: 40%
5_min_avg: 22%
15_min_avg: 10%

CPU Utilization

Usage: 25%

CPU Temp

RAM Processes

PID	USER	MEM%	RSS	VSZ	CMD
2782	user1	2.2	232332	1118240	nginx
2623	user1	2	209732	1321800	nginx
1558	user1	1.3	140796	1321400	gnome-software
5026	user1	1.2	126312	875072	nginx
5051	user1	1.1	112776	872512	nginx
1477	user1	1	108248	1471548	compiz
2997	user1	1	107208	869076	nginx

CPU Processes

PID	USER	CPU%	RSS	VSZ	CMD
5026	user1	25.5	137944	907236	nginx
2623	user1	9.5	229016	1327948	nginx
2700	user1	8.8	108324	474796	nginx
4248	user1	8.8	98848	1151460	sublime_text
1477	user1	8.6	111252	1476524	compiz
1571	user1	7.7	62984	551884	orca
1095	root	6.2	86048	457112	Xorg

Disk Partitions

NAME	STATS	USED	MOUNT
udev	0 / 4.9 G	0%	/dev
/dev/sda1	224.5 G / 449 G	50%	/
tmpfs	4.0 K / 5.0 M	1%	/run/lock

Swap Usage

FILENAME	TYPE	SIZE	USED	PRIORITY
/dev/sda5	partition	10384924	0	-1

Docker Processes

4. 实现可视化系统监控

4.1 Linux-Dash



使用Linux-Dash实现可视化实时监控

任务目标:

- 实现Linux-Dash软件的部署。
- 实现对主机系统运行情况的监控。

操作演示:



操作步骤:

- 基于PHP环境进行部署
 - 实现Linux+Apache Http Server+PHP环境
 - 部署Linux-Dash程序
 - 发布并验证
- 基于Go环境进行部署
 - 实现Go环境
 - 部署Linux-Dash程序
 - 发布并验证



4. 实现可视化系统监控

- Monitorix is a **free, open source, lightweight system monitoring tool** designed to **monitor** as many **services and system resources** as possible.
 - It has been created to be used under production Linux/UNIX servers,
 - but due to its simplicity and small size can be used on embedded devices as well.
- Monitorix 是一个开源免费、轻量级的系统监视工具。
 - 该工具主要包括两个程序：
 - 一个是自动收集Perl服务程序为monitorix;
 - 另一个为CGI脚本程序为monitorix.cgi。从Monitorix 3.0版本以后，该工具已经内置HTTP服务器（占用端口为TCP 8080），安装后可自动发布程序进行访问。



4. 实现可视化系统监控

- Monitorix 在系统运行的过程中会有规律地收集系统和网络的信息并以图形化的形式展示出来。
 - 主要特性如下所示。
 - 能够监测系统多种类型的运行状态，如CPU负载、内存使用、磁盘驱动器的温度和健康情况、网络流量、Web服务器、数据库以及系统多种服务等（如SSH、FTP、SMTP、POP3、IMAP、POP3 等）。
 - 能够按照每天、每周、每月或每年，通过图形或明文表格查看统计数字。
 - 能够缩放图形，以便更清楚地查看数据信息。
 - 内置的HTTP服务器，程序安装运行后通过Web方式访问。



4. 实现可视化系统监控



使用Monitorix实现可视化系统监控

任务目标:

- 实现Monitorix软件的部署。
- 实现对主机系统运行情况的监控。

操作步骤:

- 部署基本环境
- 安装Monitorix
- 配置监控指标
- 验证并交付
- 阅读监控数据

操作演示:



监控数据查看方式

CLI工具查看数据

可视化监控：客户端

可视化监控：Web

监控数据存储方式

实时监控（秒级呈现）

系统监控（长期存储）



网络与信息系统智能运维 课程体系学习平台

本课程体系由
河南中医药大学信息技术学院建设

课程体系学习平台由河南中医药大学医疗健康信息
工程技术研究所开发与技术保障

网络与信息系统智能运维课程体系学习平台
<https://internet.hactcm.edu.cn>

互联网运维管理工程应用丛书
<http://www.51xueweb.cn>



扫码学习
并获取课程资源

