

人工智能边缘设备操作系统



人工智能边缘设备的操作系统将涉及硬件底层的控制等关键部分进行封装,为开发人员提供一个通用的、相对简单和能驱动各种硬件工作的软件接口,同时为普通用户提供了一个操作和使用边缘设备的方法。

【模块描述】

本模块首先简要介绍操作系统基础知识,包括操作系统的概念、基本结构、分类等,介绍 常用的操作系统,引入 Linux 操作系统。其次介绍什么是 Linux 操作系统,Linux 操作系统 的特点、历史和体系结构。最后,通过在人工智能边缘设备中使用 Linux 系统的命令指令, 编写自动化脚本实现磁盘存储空间管理,从而进一步了解人工智能边缘设备操作系统的使 用方法,为后面模块的学习准备打下基础。

【学习目标】

 (1)了解设备操作系统的 (1)能够阐述和介绍操作系统的 (1)理解边缘设备操作系统的基 概念。 (2)了解边缘设备操作系统的 (2)能够使用 Linux系统基础操 系统观。 (2)从《"十四五"国家信息化规 	知识目标	能力目标	素质目标
(3) 熟悉 Linux 系统中常用的 (3) 能熟练使用 Linux 系统的文 感受智能操作系统在国家战略中	 (1) 了 解 设 备 操 作 系 统 的 概念。 (2) 了解边缘设备操作系统的特点。 (3) 熟悉 Linux 系统中常用的 	 (1)能够阐述和介绍操作系统的概念。 (2)能够使用 Linux 系统基础操作指令。 (3)能熟练使用 Linux 系统的文 	 (1)理解边缘设备操作系统的基本 概念、原理和实现技术,建立初步的 系统观。 (2)从《"十四五"国家信息化规划》 感受智能操作系统在国家战略中的

【课程思政】

介绍国产操作系统的发展历程和优势,例如,自主创新的技术路线、安全可控的特性等,以增强学生的民族自信心和自豪感。强调国产操作系统在国家信息安全和国家战略中的重要性,引导学生关注国产操作系统的研发和应用,激发他们的爱国热情和自主创新精神。

【知识框架】



【知识准备】

3.1 操作系统的概念

由于硬件设备只能处理如"01110110"这样的二进制信息,如果要将控制硬件设备的指 令命名为二进制指令,这样复杂、多变且相似的指令代码难以再被使用,因此人们在应用程 序和硬件设备之间设计了一套软件系统,这套软件系统向下可以管理和控制硬件设备,向上 可以体现为一个可视化的程序操作界面。这套软件系统也被称为操作系统。

人工智能边缘设备操作系统是在设备上加载的第一层次的软件,其他各种各样的人工 智能应用程序都是在边缘设备操作系统的支撑下完成各种特定的功能,同时也是用户与设



备底层原理的接口,能够有效管理和控制计算机系 统的软硬件资源,为用户提供操作硬件设备的平台 和方法,方便用户开发和使用人工智能应用程序。 一个典型的人工智能边缘设备系统的体系架构如 图 3-1 所示。

形式上,操作系统为用户提供一个能够与人工 智能边缘设备的系统交互的操作界面。本质上,操 作系统通过处理如设备内存管理与配置、系统资源 供需的优先次序、输入与输出设备控制、网络与文件 系统修改调整等基本事务,实现设备的软硬件资源 的管理和控制。

3.1.1 操作系统的基本功能

操作系统的类型非常多样,许多操作系统的开发者对它涵盖范畴的定义也不尽一致,例 如,有些操作系统集成了图形用户界面,而有些仅使用命令行作为系统界面,将图形用户界

面视为一种非必要的应用程序。

尽管操作系统的设计风格、用法不尽相同,但操作系统的主要功能都基本相似,包括进 程管理、内存管理、文件系统、网络通信、安全机制、用户界面和驱动程序等。一个标准的人 工智能边缘设备中的操作系统具备以下的功能。

1. 进程管理

进程管理指的是操作系统调整多个进程的功能。对于人工智能边缘设备操作系统而 言,即使只拥有一个 CPU,也可以利用多进程功能同时执行多个进程。

2. 内存管理

操作系统的存储器管理为设备提供查找可用的记忆空间、配置与释放记忆空间以及交换存储器和低速存储设备数据等功能。这种管理方法同时又被称作虚拟内存管理。

3. 文件管理

通常指磁盘文件管理,可将资料以目录或文件的形式存储。每个文件系统都有自己的 特殊格式与功能,例如,日志管理或磁盘重整。

4. 网络通信

现代的操作系统都具备操作主流网络通信协议 TCP/IP 的能力。也就是说,这样的操 作系统可以进入网络世界,并且与其他系统分享诸如文件、打印机与扫描仪等资源。

5. 安全机制

大多数操作系统都含有某种程度的信息安全机制。信息安全机制主要基于两大理念:操作系统提供外界直接或间接访问数种资源的通道和方式;操作系统有能力认证资源访问的请求。

6. 用户界面

将图形用户界面与操作系统内核紧密结合,可以提供较快速的图像回应能力,但图形系统崩溃也将导致整个系统的崩溃。因此,现代操作系统会设法将图形接口的子系统与内核分离。

7. 驱动程序

驱动程序是指某类设计用来与硬件交互的计算机软件。通常是一个设计完善的设备交 互接口,利用与此硬件连接的计算机汇排流或通信子系统,提供对此设备下命令与接收信息 的功能。

3.1.2 常见的边缘智能操作系统

由于操作系统的应用场景、性能、特点各有不同,导致还没有一个明确的分类标准为操 作系统清晰地分门别类。目前常见的分类方式有以下三种。

(1)根据系统工作方式的区别,可以分为批处理操作系统、分时操作系统、实时操作系统、网络操作系统和分布式操作系统等。

(2)根据系统架构的区别,可以分为单内核操作系统等;根据系统运行环境的区别,可以分为桌面操作系统、嵌入式操作系统等。

(3) 根据系统中使用的指令的长度,可以分为8位、16位、32位、64位的操作系统。

运行在人工智能边缘设备上的操作系统往往也有多种选择,典型的有以下一些操作 系统。

1. Linux

44

Linux 是一套基于 GNU 和 GPL 声明的免费开源和自由传播的类 UNIX 操作系统,是 一个基于 POSIX (Portable Operation System Interface of UNIX)和 UNIX 的多用户、多任 务、支持多线程和多 CPU 的操作系统。它是一个真正的多用户、多任务操作系统,具有良 好的兼容性、高度的稳定性和强大的可移植性,具有高效的开发环境和世界公认最好的语言 编辑器。

Linux 系统的特点可以概括成以下几点。

(1) 自由开放和免费性。

Linux 是一种免费的、自由的操作系统软件,获得 Linux 操作系统非常方便,大部分可 以免费从网络上下载。它是开放源代码的,爱好者可以根据自己的需要自由修改、复制和在 Internet 上发布程序源码,使用者不用担心不公开源码的系统预留"后门"。

(2) 可靠的系统安全性。

Linux采用了对读写进行权限控制、带保护的子系统、审计跟踪、核心授权等措施保证 系统安全。Linux是基于开放标准与开放源代码的操作系统,提供了更多的错误发现和修 正机制。开源操作系统可以让操作者知道问题在哪里,主动去修补漏洞,而不是被动等待软 件厂商的公告。

(3) 极好的多平台性。

虽然 Linux 主要在 x86 平台上运行,但 Linux 也能在其他主流体系结构平台上运行, Linux 是目前支持最多硬件平台的操作系统。

(4) 极高的系统稳定性。

UNIX 操作系统的稳定性是众所周知的,Linux 是基于 UNIX 规范而开发的类 UNIX 操作系统,完全符合 POSIX 标准,具有与 UNIX 相似的程序接口和操作方式,继承了 UNIX 稳定、高效、安全等特点。安装了 Linux 的主机的连续运行时间通常以年计算,系统连续运行很长的时间都不会死机,也不会出现 Windows 的蓝屏现象。目前世界上许多大型服务器 都以 Linux 作为首选的操作系统。

(5) 真正的多用户多任务。

目前虽然许多操作系统支持多任务,但只有少数的操作系统能提供真正的多任务能力。 Linux 充分利用任务切换和管理机制,是真正意义上的多用户、多任务操作系统,允许多个 用户同时执行不同的程序,且能给紧急的任务安排较高的优先级。

(6) 友好的用户界面。

Linux 同时具有字符界面和图形界面。在字符界面,用户可以通过键盘输入相应的指

令来高效地进行操作。同时,Linux还提供了类似 Windows图形界面的 X-window系统,用户可以使用鼠标方便、直观、快捷地进行操作。Linux图形界面技术已经十分成熟,其强大的功能和灵活的配置界面毫不逊色于 Windows。

(7) 强大的软件开发支持。

Linux 支持一系列的软件开发,它是一个完整的软件开发平台,支持几乎所有主流的程序设计语言。

2. DuerOS

DuerOS 是百度度秘事业部研发的对话式人工智能操作系统。作为一款开放式的操作 系统, DuerOS 能够时时进行自动学习,让机器具备人类的语言能力。简单地说,目前的 DuerOS 是面向语音交互的 AI 系统。

DuerOS 的整体架构包括三层:中间层为核心层,即对话服务系统;最上层为应用层,即 智能设备开放平台;最底层为能力层,即技能开放平台。

搭载 DuerOS 的设备可让用户以自然语言进行对话交互,实现影音娱乐、信息查询、生活服务、出行路况等多项功能的操作,同时支持第三方开发者的能力接入。DuerOS 整体技术架构如图 3-2 所示。

	参考设计							
应用层		开发套件			Ţ	结音		
智能设备开放平台	核心	核心接入组件			业设	构 腔 设		
	设备唤醒 SD	K 系统适配层 安全	上	阵列	计	में में		
			对话服务					
核心层	语音识	语音识别		语音识别				
对话核心系统	自然语言理解	自然语言理解 对话状态控制			吾言生成	搜索		
	知识图谱	知识图谱 网页图谱			言息图谱	用户画像		
	Į.	原生技能		第	三方技能			
能力层	音乐 电视器	割 咨询 天空	气 购物	勿 打车	三 新闻	休闲		
技能开放平台	电影 短视	频 问答 …	… 聊え	モ 电台	; 游戏	· · · · · · · · · · · · · · · · · · ·		
	技能开发工具							

图 3-2 DuerOS 整体技术架构

3. AliOS-Things

AliOS-Things 是由阿里巴巴研发的一款轻量级嵌入式操作系统。AliOS-Things 适配性广,支持多种 CPU 架构,包括 ARM、C-Sky、MIPS、rl78、rx600、Xtensa 等。

AliOS-Things 支持接入多种物联网平台,包括阿里巴巴智能天猫精灵、亚马逊云、

Google Home 音箱控制等。同时,AliOS-Things 支持接入阿里巴巴的飞燕平台,具备设备 开发调试、设备加密、云端开发、App 开发、运营管理、数据统计等功能,能够很好地覆盖从 前期开发到后期运营的人工智能边缘设备应用开发全生命周期。AliOS-Things 主要具有 如下特点。

(1) 支持多种语言开发,包括C、JavaScript等语言。

(2)提供集成开发环境,支持代码编辑、编译、调试、内存泄漏检测等。

(3) 支持 App 独立升级。

(4) 支持组件式开发。

AliOS-Things 技术架构如图 3-3 所示。



图 3-3 AliOS-Things 技术架构

3.2 Linux 系统操作命令

Linux 系统中有些图形化工具极大地降低了运维人员操作出错的概率,但是,很多图形 化工具其实是调用了脚本来完成相应的工作,往往只是为了完成某种工作而设计的,缺乏 Linux 命令原有的灵活性及可控性。再者,图形化工具相较于 Linux 命令行界面会更加消 耗系统资源,因此有些经验丰富的边缘智能应用开发人员甚至都不会给 Linux 系统安装图 形界面,仅在工作时直接通过命令行模式远程连接人工智能边缘设备进行开发和调试。

Shell 就是这样的一个命令行工具。Shell 也称为终端或壳,其充当的是用户与硬件设备之间翻译官的角色,用户把一些命令"告诉"操作系统,它就会调用相应的程序服务去完成某些工作。现在包括红帽系统在内的许多主流 Linux 系统默认使用的终端是 Bash (Bourne-Again SHell) 解释器。

3.2.1 系统状态检测命令

作为一名合格的人工智能边缘设备开发和运维人员,想要更快、更好地了解 Linux 服务器,必须具备快速查看 Linux 系统运行状态的能力,因此接下来会逐个讲解与网卡网络、系统内核、系统负载、内存使用情况、当前启用终端数量、历史登录记录、命令执行记录以及救援诊断等相关命令的使用方法。

1. ifconfig 命令

ifconfig 命令用于获取网卡配置与网络状态等信息,格式为"ifconfig [网络设备][参数]"。

使用 if config 命令来查看本机当前的网卡配置与网络状态等信息时,其实主要查看的 就是网卡名称、inet 参数后面的 IP 地址、ether 参数后面的网卡物理地址(又称为 MAC 地址),以及 RX、TX 的接收数据包与发送数据包的个数及累计流量。

在 Shell 中输入命令"ifconfig",输出如下。

```
[root@tringai ~]#ifconfig
eno16777728: flags=4163 mtu 1500
       inet 192.168.10.10 netmask 255.255.255.0 broadcast 192.168.10.255
       inet6 fe80::20c:29ff:fec4:a409 prefixlen 64 scopeid 0x20
       ether 00:0c:29:c4:a4:09 txqueuelen 1000 (Ethernet)
       RX packets 36 bytes 3176 (3.1 KiB)
       RX errors 0 dropped 0 overruns 0 frame 0
       TX packets 38 bytes 4757 (4.6 KiB)
       TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0
lo: flags=73 mtu 65536
       inet 127.0.0.1 netmask 255.0.0.0
       inet6::1 prefixlen 128 scopeid 0x10
       loop txqueuelen 0 (Local Loopback)
       RX packets 386 bytes 32780 (32.0 KiB)
       RX errors 0 dropped 0 overruns 0 frame 0
       TX packets 386 bytes 32780 (32.0 KiB)
       TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0
```

2. uptime 命令

uptime 用于查看系统的负载信息,格式为 uptime。

uptime 命令可以显示当前系统时间、系统已运行时间、启用终端数量以及平均负载值 等信息。平均负载值指的是系统在最近 1min、5min、15min 内的压力情况;负载值越低越 好,尽量不要长期超过 1min,在生产环境中不要超过 5min。

在 Shell 中输入命令"uptime",输出如下。

```
[root@tringai ~]#uptime
22:49:55 up 10 min, 2 users, load average: 0.01, 0.19, 0.18
```

上述输出信息中的"0.01""0.19""0.18"分别对应该系统在最近 1min、5min、15min 内的 压力情况。

3. free 命令

48

free 用于显示当前系统中内存的使用量信息,格式为"free [-h]"。

为了保证 Linux 系统不会因资源耗尽而突然宕机,运维人员需要时刻关注内存的使用 量。在使用 free 命令时,可以结合使用-h 参数以更人性化的方式输出当前内存的实时使用 量信息。

在 Shell 中输入命令"free -h",输出如表 3-1 所示。

```
[root@tringai ~]#free -h
```

字 段 名	内存总量	已用量	可用量	进程共享 的内存量	磁盘缓存 的内存量	缓存的 内存量
total	used	free	shared	buffers	cached	
Mem:	1.8GB	1.3GB	542MB	9.8MB	1.6MB	413MB
-/+ buffers/cache:		869MB	957 MB			
Swap:	2.0GB	0B	2.0GB			

表 3-1 "free -h"输出结果

4. history

history 命令用于显示历史执行过的命令,格式为"history [-c]"。

执行 history 命令能显示出当前用户在本地计算机中执行过的最近 1000 条命令记录。如果觉得 1000 条不够用,还可以自定义/etc/profile 文件中的"HISTSIZE"变量值。在使用 history 命令时,如果使用-c 参数则会清空所有的历史命令记录。还可以使用"! 编码数字"的方式来重复执行某一次的命令。

在 Shell 中输入命令"history",输出如下。

```
[root@tringai ~]#history
1 ipconfig
2 uptime
3 free
4.history
```

5. df 命令

df 命令用于显示 Linux 系统中各文件系统的磁盘使用情况,包括文件系统所在磁盘分区的总容量、已使用的容量、剩余容量等。df 命令主要读取的数据几乎都针对的是整个文件

系统,其作用主要是从各文件系统的 Super block 中读取数据。

df 命令的基本格式为

```
[root@tringai~]#df [选项] [目录或文件名]
```

df 命令常用的几个选项及其作用如表 3-2 所示。

表 3-2 df 命令常用的选项及其作用

选 项	作用
-a	显示所有文件系统信息,包括系统特有的 /proc、/sysfs 等文件系统
-m	以 MB 为单位显示容量
-k	以 KB 为单位显示容量,默认以 KB 为单位
-h	使用人们习惯的 KB、MB 或 GB 等单位自行显示容量
-T	显示该分区的文件系统名称
-i	不用磁盘容量显示,而是以含有 inode 的数量来显示

3.2.2 工作目录切换命令

工作目录指的是用户当前在系统中所处的位置,常用的命令如下。

1. pwd 命令

pwd 命令用于显示用户当前所处的工作目录,格式为"pwd [选项]"。 在 Shell 中输入命令"pwd",输出如下。

```
[root@tringai etc]#pwd
/etc
```

2. cd 命令

cd命令用于切换工作路径,格式为"cd [目录名称]"。

这是一个较为常用的 Linux 命令。可以通过 cd 命令迅速、灵活地切换到不同的工作目录。除了常见的切换目录方式,还可以使用"cd-"命令返回到上一次所处的目录,使用 "cd.."命令进入上级目录,以及使用"cd~"命令切换到当前用户的主目录。例如,可以使用 "cd 路径"的方式切换进/etc 目录中。

[root@tringai ~]#cd /etc

同样的道理,可使用下述命令切换到/bin 目录中。

[root@tringai etc]#cd /bin

此时,要返回到上一次的目录(即/etc目录),可执行如下命令。

```
[root@tringai bin]#cd -
/etc
[root@tringai etc]#
```

还可以通过下面的命令快速切换到用户的主目录。

```
[root@tringai etc]#cd ~
[root@tringai ~]#
```

3. ls 命令

50

ls命令用于显示目录中的文件信息,格式为"ls[选项][文件]"。

所处的工作目录不同,当前工作目录下的文件肯定也不同。使用 ls 命令的"-a"参数可 以看到全部文件(包括隐藏文件),使用"-l"参数可以查看文件的属性、大小等详细信息。将 这两个参数整合之后,再执行 ls 命令即可查看当前目录中的所有文件并输出这些文件的属 性信息。

```
[root@tringai ~]#ls -al
total 60
dr-xr-x---. 14 root root 4096 May 4 07:56 .
drwxr-xr-x. 17 root root 4096 May 4 15:55 ..
-rw-----. 1 root root 1213 May 4 15:44 anaconda-ks.cfg
-rw-----. 1 root root 957 May 4 07:54 .bash history
```

3.2.3 文件目录管理命令

在 Linux 系统的日常运维工作中,还需要掌握对文件的创建、复制、更名与删除等文件 目录管理操作。常用以下命令进行文件目录管理。

1. touch 命令

touch 命令用于创建空白文件或设置文件的时间,创建空白文件时,格式为"touch [选项] [文件]"。

在 Shell 中输入命令"touch",输出如下。

[root@tringai~]#touch file #创建一个名为"file"的新的空白文件

对 touch 命令来讲,有难度的操作主要是体现在设置文件内容的修改时间(mtime)、文件权限或属性的更改时间(ctime)与文件的读取时间(atime)上面。touch 命令的参数及其作用如表 3-3 所示。

参数	作用
-a	仅修改"读取时间"(atime)

表 3-3 touch 命令参数介绍

续表

参数	作用
-m	仅修改"修改时间"(mtime)
-d	同时修改 atime 与 mtime

接下来,先使用 ls 命令查看一个文件的修改时间,然后修改这个文件,最后再通过 touch 命令把修改后的文件时间设置成修改之前的时间。

```
[root@tringai ~]#ls -l anaconda-ks.cfg
-rw-----. 1 root root 1213 May 4 15:44 anaconda-ks.cfg
[root@tringai ~]#echo "Visit the tringai.com to learn linux skills" >>anaconda
-ks.cfg
[root@tringai ~]#ls -l anaconda-ks.cfg
-rw-----. 1 root root 1260 Aug 2 01:26 anaconda-ks.cfg
[root@tringai ~]#touch -d "2017-05-04 15:44" anaconda-ks.cfg
[root@tringai ~]#ls -l anaconda-ks.cfg
-rw-----. 1 root root 1260 May 4 15:44 anaconda-ks.cfg
```

2. mkdir 命令

mkdir 命令用于创建空白的目录,格式为"mkdir [选项] 目录"。

在 Linux 系统中,文件夹是最常见的文件类型之一。除了能创建单个空白目录外, mkdir 命令还可以结合-p 参数来递归创建出具有嵌套叠层关系的文件目录。

```
[root@tringai ~]#mkdir tringai
[root@tringai ~]#cd tringai
[root@tringai tringai]#mkdir -p a/b/c/d/e
[root@tringai tringai]#cd a
[root@tringai a]#cd b
[root@tringai b]#
```

3. cp 命令

cp 命令用于复制文件或目录,格式为"cp [选项] 源文件 目标文件"。在 Linux 系统中, 复制操作具体分为以下三种情况。

(1) 如果目标文件是目录,则会把源文件复制到该目录中。

(2) 如果目标文件也是普通文件,则会询问是否要覆盖它。

(3) 如果目标文件不存在,则执行正常的复制操作。

cp命令的参数及其作用如表 3-4 所示。

接下来,使用 touch 命令创建一个名为 install.log 的普通空白文件,然后将其复制为一个名为 x.log 的备份文件,最后再使用 ls 命令查看目录中的文件。

参数	作用
-р	保留原始文件的属性
-d	若对象为"链接文件",则保留该"链接文件"的属性
-r	递归持续复制(用于目录)
-i	若目标文件存在则询问是否覆盖
-a	相当于-pdr(p、d、r为上述参数)

表 3-4 cp 命令参数介绍

```
[root@tringai ~]#touch install.log
[root@tringai ~]#cp install.log x.log
[root@tringai ~]#ls
install.log x.log
```

4. mv 命令

mv命令用于剪切文件或将文件重命名,格式为"mv[选项]源文件[目标路径|目标文件名]"。

剪切操作不同于复制操作,因为它会默认把源文件删除掉,只保留剪切后的文件。如果 在同一个目录中对一个文件进行剪切操作,其实也就是对其进行重命名。

```
[root@tringai ~]#mv x.log linux.log
[root@tringai ~]#ls
install.log linux.log
```

5. rm 命令

rm 命令用于删除文件或目录,格式为"rm [选项] 文件"。

在 Linux 系统中删除文件时,系统会默认向用户询问是否要执行删除操作,如果不想总 是看到这种反复的确认信息,可在 rm 命令后加上-f 参数来强制删除。另外,如果想要删除 一个目录,需要在 rm 命令后面加-r 参数才可以,否则无法删除。接下来尝试删除前面创建 的 install.log 和 linux.log 文件。

```
[root@tringai ~]#rm install.log
rm: remove regular empty file 'install.log'?y
[root@tringai ~]#rm -f linux.log
[root@tringai ~]#ls
[root@tringai ~]#
```



【案例实现】 Linux 系统磁盘检测与维护

在计算机领域,日志文件(logfile)是一个记录了发生在运行中的操作系统或其他软件

中的事件的文件,或者记录了在网络聊天软件的用户之间发送的消息。人工智能边缘设备 在运行 AI 程序过程中会产生大量日志文件占用磁盘空间。但是磁盘空间有限,所以磁盘 常常很快就处于用完状态,并由此产生一些问题,如 AI 应用运行过程中由于存储空间不足 的问题导致终端挂掉,进而影响生产应用。

在本项目实训中,将首先了解如何使用常用的设备运维工具检查设备情况,接着了解如 何手动查看磁盘存储空间情况,最后通过创建自动化脚本的方式自动删除系统日志。

本次案例实训的思路如下。

(1)使用 jtop 工具查看设备情况。人工智能边缘设备作为新一代自主机器设计的嵌入 式系统,是一个基于 Jetson 系列芯片搭建的人工智能硬件平台,具备 CPU、GPU、PMIC、 DRAM、闪存以及可扩展性。Jetson 中提供了设备自动监控工具 jtop。

(2) 手动查看具体磁盘存储空间。通过命令行的方式,分辨文件类型,查看文件存储 情况。

(3)自动删除磁盘存储日志文件。通过自动化脚本的方式,编写自动清理日志文件脚本,并定时执行。

任务 1: 使用 jtop 工具查看设备情况

首先双击桌面图标 Terminal 或按 Ctrl+Alt+T 组合键,打开 Linux 系统终端,如图 3-4 所示。



图 3-4 打开 Linux 系统终端

通过以下命令安装 jtop 工具。

sudo -H pip install -U jetson-stats

1. 查看设备状态

安装完成后,通过以下命令使用 jtop 工具。

jtop

输出结果如图 3-5 所示。

🙁 🗖 🗊 jtop Nano - JC: Inactive - M	IAXN				
R I	top Nano - JC:	Inactive - MAXN	80x24		
NVIDIA Jetson Nano - Jetpack CPUI [Schedutil - 1 CPUZ [Schedutil - 1 CPU3 [Schedutil - 1 CPU4 [Schedutil - 1	4.5.1 [L4T .2%] 1.2GHz .0%] 1.2GHz .4%] 1.2GHz .1%] 1.2GHz	32.5.1]			
Mem [[] (lfb kB] (lf B] (cac 2%	299x4MB) b 252kB) hed 0MB)] 1.6GHz
GPU [Dsk [####################################	- [Sensor] - AO CPU GPU PLL thermal	— [Temp] 53.00C 47.50C 45.00C 45.50C 46.00C	[Power/mW] - SV CPU SV GPU ALL	0% 37.4GB/ [Cur] 297 42 2291] 76 MHz 116.9GB] [Avr]
1ALL 2GPU 3CPU 4MEM 5CTF	L GINFO (Juit		Raffael	lo Bonghi

图 3-5 查看设备状态

可以看到,jtop 工具中主要包括 ALL、GPU、CPU、MEM、CTRL、INFO 这 6 个子页面。 使用鼠标单击对应子页面或使用键盘输入子页面序号即可进行页面切换。

在 ALL 页面中,可以查看包含模块运行信息: CPU 使用情况、内存使用情况、GPU 使用情况、风扇状态等。同时可以看到设备的 Jetpack 版本号。Jetpack 是一个用于快速安装和搭建人工智能应用程序的工具包,可以帮助开发者快速搭建人工智能程序运行所需要的环境,版本的升级迭代会带来功能服务的差异。

2. 查看 GPU 状态

切换到"2GPU"页面中,可以实时查看 GPU 的使用状态,如图 3-6 所示。



图 3-6 实时查看 GPU 的使用状态

其中,Jetson Clocks 是 Jetson 自带的工具,用于自动动态调整设备的 GPU 使用,使得处理器运行在最高性能。设备默认是不开启 Jetson Clocks 模式的,也就是一个平衡性能的状态,"Jetson Clocks:inactive"表示该动态调整已关闭,设备正处于平衡性能的状态。如需要使用最大 GPU 性能,可以通过以下命令进行开启。

sudo jetson clocks

"NV Power[0]: MAXN"表示设备当前状态为不使用 GPU 的最大功率模式,为 10W。

3. 查看 CPU 状态

Jetpack 4.5.1 [L4T 32.5.1] VIDIA Jetson Nano -CPU1 45% Schedutil CPU2 43% Schedutil Platform Arch: aarch64 100% Rel: 4.9.201-tegra 87% Cluster: 75% 62% PU1 ARMV8 50% Frq: 1.5GHz 37% 25% ARMv8 Frq: 1.5GHz 12% ARMv8 0% Frq: 1.5GHz CPU3 51% Schedutil CPU4 47% Schedutil Frq: 1.5GHz 100% 87% 75% 62% 50% 37% 25% 12% 0% -2s 4s -45 65 time 3CPU 4MEM 5CTRL Raffaello Bonghi 2GPU 6INFO Ouit 1ALL

切换到"3CPU"页面中,可以实时查看 CPU 的使用状态,如图 3-7 所示。

图 3-7 实时查看 CPU 的使用状态

在 CPU 页面右侧,可以看到 4 颗 CPU 的实时运行状态。在 CPU 页面左侧,有一列 "Platform",主要显示设备的架构以及 CPU 的运行情况。其中,"Arch: aarch64"表示设备 为 ARMv8-A 架构中引入的 64 位状态。从图中可以看到,人工智能边缘设备中包含编号为 1~4 在内的共 4 颗 CPU,"Frq: 1.5GHz"表示当前 CPU 主频为 1.5GHz,即 CPU 内核工作 的时钟频率,在一定程度上反映 CPU 速度的快慢。

4. 查看 MEM 页面

切换到"4MEM"页面中,可以查看设备磁盘空间的使用状态,如图 3-8 所示。

在该页面中有以下三个常用功能。

(1)清除缓存。按C键可实现快速清除设备缓存。

(2) 启用或禁用额外交换。人工智能模型训练过程会涉及大量的数据,通常在使用过 程中需要占用大量空间,当出现运行空间不足以支持人工智能程序运行时,可以通过将设备 磁盘存储空间分配为程序运行空间的方式以保证程序运行。按S键可实现启用或禁用运行 空间交换。

(3) 增加和减少交换大小。按"+"键可实现增加虚拟运行空间,按"-"键可实现减少

NVIDIA Jetson Nano - Jetpack 4.5.1 [L4T RAM 1.5G/4.1GB - (lfb 308x4MB)	32.5.1] Swap - 4.1G - 3.6G - 3.1G - 2.6G - 2.0G	(cached 0MB) [0/519.6Mb] P= 5 [0/519.6Mb] P= 5 [0/519.6Mb] P= 5 [0/519.6Mb] P= 5
-6s -4s -2s	- 1.5G TOT [- 1.0G - 0.5G - 0.0G 0 time	0.0GB/2.0GB]
Mem [1.5G/4.1GB] (lfb 308x4MB) 0.0k/252.0kB] (lfb 252kB) 2%] 1.6GHz
<u>c</u> Clear cache <u>Extra Status</u>		RAM Legend CPU: 1.4 GB GPU: 141.9 MB
Swap Disable - 2 40 1ALL 2GPU 3CPU 4MEM 5CTRL 6INFO	Quit	Raffaello Bonghi

图 3-8 查看设备磁盘空间的使用状态

虚拟运行空间。

5. 查看 CTRL 页面

切换到"5CTRL"页面中,可以对设备的性能进行控制,如图 3-9 所示。

NVIDIA Jetson Nano - Jetpack	4.5.1 [L4	T 32.5.1]				
Fan Fan Speed <u>n</u> 0% <u>p</u> Mode	defau	lt sys	stem r	nanual		
<u>s</u> <u>jetson_clocks</u> Inactive		FAN	0% of	0% Auto	=Enable	- 100% - 93%
<u>e</u> <u>boot</u> Disable						- 80% - 73% - 66%
NVP model 0 +						- 60% - 53%
MAXN 5W						- 46% - 40% - 33%
						- 26% - 20%
						- 13% - 6% - 0%
			-6s	-4s	-2s	0 time
1ALL 2GPU 3CPU 4MEM 5CTR	RL 6INFO	Quit			Raffael	lo Bonghi.

图 3-9 设备性能控制界面

在该页面中有如下常用功能。

(1) 控制风扇运行。按 P 键可实现提高设备风扇运行速度,加快散热,保证设备计算运 行效率;按 M 键可实现降低设备风扇运行速度,降低能耗。

(2) 控制 GPU 的使用数量。按"+"键表示使用一颗 GPU,按"-"键表示不使用 GPU。另外,在使用一颗 GPU 的模式下,设备默认功率为 5W。

任务2:手动查看具体磁盘存储空间

jtop 工具中为运维人员查看设备整体情况提供了便利,但无法查看具体某一任务下的执行情况。接下来以查看系统日志文件为例,学习如何查看磁盘具体路径下的文件存储情况,在终端中输入如下命令,查看设备当前存储空间情况。

df -h

如图 3-10 所示,输出结果中,第一列是文件系统,第二列是该文件系统所拥有的可支配 空间,第三列是已用空间,第四列是可用空间,第五列是目前该文件系统的可支配空间的利 用率,第六列是文件系统所处位置。

tringai@tringai	:~\$ df	- h			
Filesystem	Size	Used	Avail	Use%	Mounted on
none	1.7G	Θ	1.7G	0%	/dev
tmpfs	2.0G	88K	2.0G	1%	/dev/shm
tmpfs	2.0G	30M	2.0G	2%	/run
tmpfs	5.0M	4.0K	5.0M	1%	/run/lock
tmpfs	2.0G	0	2.0G	0%	/sys/fs/cgroup
/dev/nvme0n1p1	117G	38G	74G	34%	
tmpfs	396M	124K	396M	1%	/run/user/1000
/dev/mmcblk0p1	14G	12G	1.3G	91%	/media/tringai/325238d3-3fd0-45ab-a8d9-c43dad4c7dda1
/dev/sda1	115 <u>G</u>	97G	18G	85%	/media/tringai/82AAE4FAAAE4EB9B

图 3-10 查看设备当前存储空间情况

本次项目将以清理系统日志文件为例,进行磁盘存储空间清理。在人工智能边缘设备 中。系统日志文件默认存放在"var/log"文件夹中,通过以下命令进入"log"文件夹中。

cd /var/log

使用常用的"ls"命令,输出结果如图 3-11 所示。

tringai@tringai:~\$ cd /var/log									
tringai@tringai:/var/log\$ ls									
alternatives.log	auth.log.1	installer	syslog.4.gz						
alternatives.log.1	auth.log.2.gz	kern.log	syslog.5.gz						
apport.log	auth.log.3.gz	kern.log.1	syslog.6.gz						
apport.log.1	auth.log.4.gz	kern.log.2.gz	syslog.7.gz						
apport.log.2.gz	btmp	kern.log.3.gz	tallylog						
apport.log.3.gz	btmp.1	kern.log.4.gz	wtmp						
apport.log.4.gz	dpkg.log	lastlog	wtmp.1						
apport.log.5.gz	dpkg.log.1	oem-config.log	x11vnc.log						
apport.log.6.gz	dpkg.log.2.gz	syslog	Xorg.0.log						
apport.log.7.gz	dpkg.log.3.gz	syslog.1	Xorg.0.log.old						
apt	fontconfig.log	syslog.2.gz	Xorg.1.log						
auth.log	gdm3	syslog.3.gz	Xorg.1.log.old						

图 3-11 "var/log"文件夹内容

输出结果中只能看到文件名、文件类型。在进行磁盘管理时,显然应该关注的是文件的 大小属性。可以通过以下命令查看当前目录下各文件所占空间大小。

du -sh

如图 3-12 所示,输出结果中表示 log 文件夹大小为 86MB。 如果想知道其中每个文件的大小,可以通过以下命令实现,如图 3-13 所示。



tringai@tringai:/var/log\$ du -sh du: cannot read directory './gdm3': Permission denied

图 3-12 查看当前目录下各文件所占空间大小

du -ah

tringa	gai@tringai:/var/log\$ du -ah		
292K	./lastlog		
0	./alternatives.log		
8.0K	./x11vnc.log		
o	./btmp.1		
8.0K	./auth.log.1		
8.0K	./auth.log.2.gz		
O	./dpkg.log		
24K	./Xorg.0.log		
4.0K	./dpkg.log.2.gz		
8.0K	./dpkg.log.1		
2.1M	./syslog.2.gz		
30M	./kern.log		
4.0K	./apt/history.log.2.gz		
4.0K	./apt/history.log.1.gz		
4.0K	./apt/term.log.2.gz		
0	./apt/history.log		
44K	./apt/term.log.3.gz		
4.0K	./apt/term.log.1.gz		
16K	./apt/history.log.3.gz		
0	./apt/term.log		
112K	./apt/eipp.log.xz		
192K	./apt		
2.1M	./svslog.3.gz		

图 3-13 当前目录下每个文件的大小

任务 3: 自动删除磁盘存储日志文件

实际工作中,部署到设备的人工智能应用可能随时在往存储空间中写入日志,有些人工 智能应用运行一次就会产生大量日志文件,在此情况下,运维人员需要每天按时去看一下磁 盘情况吗?显然这会很麻烦,接下来编写 Linux 定时任务脚本,定时运行并自动删除冗余的 系统日志文件。

1. 创建自动执行脚本

首先创建一个脚本文件,具体命令如下。

```
touch auto_sel_log.sh
ls
```

如图 3-14 所示,可以看到桌面目录下新增了一个名为 auto_sel_log.sh 的普通文件。 接着为该脚本文件赋予执行权限。具体命令如下。

```
chmod +x auto_sel_log.sh
ls
```

如图 3-15 所示,执行完后,可以看到 auto_sel_log.sh 文件名变色,表示此时文件变为可执行文件。若再使用 chmod -x 命令,则可以去除该文件的执行权限。

接着编辑刚创建的脚本文件,具体代码如下。



tringai@tringai:~\$ touch auto_sel_lo tringai@tringai:~\$ ls	g.sh
auto_sel_log.sh	nvidia-vpi_demos-1.0.desktop
ch343ser_linux(CH9102的linux驱动)	nv_jetson_zoo.desktop
Desktop	Pictures
Documents	Public
Downloads	rootOnNVMe-master
examples.desktop	ROS常用功能命令.txt
imu.sh	Templates
jetson-inference	tringai_qt.sh
jetson-voice	tringai_robot
Music	Videos
nv_devzone.desktop	VisionWorks-SFM-0.90-Samples
nv_forums.desktop	yolo在NX-jetpack4.5上安装

图 3-14 创建 auto_sel_log.sh 文件

<pre>tringai@tringai:~\$ chmod +x auto_sel</pre>	_log.sh
tringai@tringai:~\$ ls	
auto_sel_log.sh	nvidia-vpi_demos-1.0.desktop
ch343ser_linux(CH9102的linux驱动)	nv_jetson_zoo.desktop
Desktop	Pictures
Documents	Public
Downloads	rootOnNVMe-master
examples.desktop	ROS常用功能命令.txt
imu.sh	Templates
jetson-inference	tringai_qt.sh
jetson-voice	tringai_robot
Music	Videos
nv_devzone.desktop	VisionWorks-SFM-0.90-Samples
nv_forums.desktop	yolo在NX-jetpack4.5上安装

图 3-15 为脚本文件赋予执行权限

vi auto_sel_log.sh

输入 i,可以看到左下角为"--INSERT--",表示当前为编辑模式。在文件中输入以下代码,表示该脚本文件可执行查询磁盘存储空间功能。

find /var/log/ -mtime +2 -name "*.log" -exec rm -rf {} \;

该代码解析如表 3-5 所示。

表 3-5 代码解析

	解 释
find	查找命令,查找指定文件路径
/var/log/	需要定时清理的文件的目录位置
-mtime	标准语句写法
+2	数字指代天数,即删除2天以前的
-name " * .log"	目标文件类型
-exec	标准语句写法
rm -rf	强制删除包括目录在内的文件
{} \;	标准语句写法

2. 定时执行脚本任务

接下来安装自动化执行工具 Crontab 帮助我们更快地完成该任务。首先进入 root 权限安装该工具,密码为 tringai。

sudo su

60

接着安装 Crontab 工具。具体命令如下。

apt-get install cron

在终端中输入以下代码,配置自动运行脚本。

crontab -e

输入 i,可以看到左下角为"--INSERT--",表示当前为编辑模式。如果希望自动化周期 性地执行编写好的程序,可以通过以下代码方式实现。

* * * * * 脚本文件

其中,前5个"*"分别表示分钟、小时、日期、月份和星期。具体数值要求如表 3-6 所示。

字 段	说 明
分钟	取值为 0~59 的任意整数
小时	取值为 0~23 的任意整数
日期	取值为1~31的任意整数
月份	取值为1~12的任意整数
星期	取值为 0~7 的任意整数,0 或 7 代表星期日
命令	要执行的命令或程序脚本

表 3-6 数值要求

在通常情况下,将设置每天或每周一次的频率进行清理。为方便看到脚本运行情况,在本项目案例中,设置每天间隔 1min 自动执行一次脚本程序。执行完 crontab -e 后进行如下 配置,如图 3-16 所示。

1 * * * * /home/tringai/Desktop/auto_sel_log.sh

接着通过以下命令启动定时任务。

service cron start

由于该自动化脚本在运行过程中不会实时输出信息,如果需要查看脚本是否正常运行, 可以通过以下命令查看脚本运行情况,如图 3-17 所示。