

LPIC1 (T1)

Ghazaleh Keyvani

2025/7/7

XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX

1. خلاصه کامپوننت‌های سرور:

- مادربرد (System Board): صفحه اصلی مدار چاپی که تمام قطعات به آن متصل می‌شوند.
- CPU: مغز پردازشی سرور (مثال: Intel Xeon یا AMD EPYC).
- RAM: حافظه موقت برای اجرای برنامه‌ها (پیکاده‌سازی: `free -h` در لینوکس).
- Storage Drives
 - HDD: دیسک چرخان مکانیکی (ارزان، ظرفیت بالا)
 - SSD: حافظه فلش (سرعت بالا، بدون قطعه متحرک)
 - NVMe: SSD با رابط PCIe (بالاترین سرعت)
- RAID Controller: کنترلر مدیریت چند دیسک (پیکاده‌سازی: RAID 0/1/5).
- PSU: منبع تغذیه (مثال: واحد 800W با قابلیت Redundancy).
- NIC: کارت شبکه (مثال: 1 Gbps یا 10 Gbps).
- Cooling System: فن‌ها و هیترسینک‌ها (مثال: خنک‌کاری مایع در سرورهای پرترافیک).
- PCIe Slots: اسلات توسعه برای کارت‌های GPU/FPGA.
- Chassis: بدنه فیزیکی (Rack: رک‌مونت، Tower: ایستاده، Blade: ماژولار).
- BIOS/UEFI: فریمور پایه‌ای برای بوت سیستم.
- Backplane: برد رابط برای اتصال دیسک‌ها به مادربرد.

2. iLO و IPMI:

- IPMI: استاندارد مدیریت خارج از باند سرور (حتی هنگام خاموشی)
- iLO: نسخه اختصاصی HP از IPMI
- کاربردها:
 - مانیتورینگ سخت‌افزار (دمای CPU، ولتاژ)

- روشن/خاموش کردن ریموت
 - دسترسی به کنسول مجازی
 - بهروزرسانی فریمور
-

3. ارتباط با BIOS/UEFI:

- IPMI/iLO می‌توانند:
 - تنظیمات BIOS/UEFI را تغییر دهند
 - فرآیند بوت را مانیتور کنند
 - بهروزرسانی فریمور انجام دهند
 - مثال: تغییر تنظیمات بوت از Legacy به UEFI از طریق iLO
-

4. CPU Sockets:

- سوکت‌های فیزیکی روی مادربرد برای نصب CPU
 - هدف:
 - پشتیبانی از ارتقاء CPU
 - تطابق با معماری خاص (مثال: سوکت LGA 3647 برای Intel Xeon)
 - پیاده‌سازی: در سرورهای چندپردازنده‌ای مثل 4 Dell PowerEdge R940 (سوکت)
-

5. فلسفه ایجاد Pseudo File System در لینوکس:

- فلسفه لینوکس: "همه چیز فایل است" (Everything is a file)
 - هدف: ارائه یکپارچه اطلاعات سیستمی از طریق رابط فایل سیستم
 - دسترسی به داده‌های کرنل
 - مدیریت دستگاه‌ها به صورت یکنواخت
 - ساده‌سازی عملیات (مثال: خواندن دمای CPU با `cat`)
-

6. تفاوت Pseudo FS و Normal FS:

Pseudo FS (مثل `/proc`, `/sys`)

Normal FS (مثل `ext4`, `NTFS`)

داده‌ها روی دیسک ذخیره می‌شوند

داده‌ها در حافظه موقت (RAM)

محتوای ایستا

محتوای پویا و لحظه‌ای

ساختار واقعی دایرکتوری/فایل

ساختار سلسله‌مراتبی مجازی

مثال: `/proc/cpuinfo`

مثال: `/home/user/file.txt`

7. اطلاعات دایرکتوری `/sys/`:

• بلوک‌های نمونه:

- `/sys/class/net/`: کارت‌های شبکه
- `/sys/devices/`: دستگاه‌های فیزیکی
- `/sys/power/`: مدیریت انرژی
- `/sys/kernel/`: تنظیمات کرنل

• پیاده‌سازی عملی:

• `bash`

```
# مشاهده مدل CPU
cat /sys/devices/cpu/model
```

```
# مشاهده سرعت فن (مثال واقعی)
```

```
cat /sys/class/hwmon/hwmon0/fan1_input
```

8. DMA (Direct Memory Access):

- مفهوم: دسترسی مستقیم دستگاه‌ها به حافظه بدون دخالت CPU
- کاربرد در لینوکس:
 - افزایش سرعت انتقال داده (مثلاً در دیسک‌ها و کارت‌های شبکه)

○ کاهش بار پردازشی CPU

- مثال: کارت SSD NVMe از DMA برای انتقال داده با سرعت 3500MB/s استفاده می‌کند

9. عملکرد دستورات `*ls`:

- `lsblk`: خواندن اطلاعات از `/sys/block/` و `/sys/devices/`
- `lsusb/lspci`: خواندن از `/sys/bus/usb/` و `/sys/bus/pci/`
- `lshw`: جمع‌آوری اطلاعات از منابع مختلف (`.sysfs`, `DMI`, `etc`)
- پیاده‌سازی آزمایشی:
- `bash`

ردگیری عملکرد `lsblk`

- `strace -e openat lsblk`

10. شبیه‌سازی Shutdown از طریق `sysfs`:

`bash`

نیاز به دسترسی روت دارد

```
echo 1 | sudo tee /sys/class/power_supply/BAT0/device/shutdown
```

توجه: مسیر دقیق بسته به سخت‌افزار متفاوت است

11. انواع کرنل‌ها:

نوع	مزایا	معایب	مثال
Monolithic (یکپارچه)	عملکرد بالا	پایداری کمتر	لینوکس
Microkernel	پایداری بالا	عملکرد پایین	MINIX
(ریزکرنل)			

12. دلیل استفاده از اولین سکتور برای MBR:

- سنت طراحی: BIOS همیشه سکتور 0 سیلندر 0 هد 0 را می‌خواند
- محدودیت تاریخی: اولین محل قابل آدرس‌دهی در دیسک‌های قدیمی

13. عملکرد MBR در بوت:

1. BIOS سکتور اول (512 بایت) را می‌خواند
2. کد اجرایی (446 MBR بایت اول) اجرا می‌شود
3. MBR پارتیشن فعال را شناسایی می‌کند
4. سکتور بوت پارتیشن فعال (حاوی bootloader) را لود می‌کند
5. GRUB مرحله 2 اجرا می‌شود

14. فایل‌های efi:

- نقش: باینری‌های اجرایی برای بوت UEFI
- محل: پارتیشن ESP (معمولاً `/boot/efi/`)
- مثال: `grubx64.efi`, `bootmgfw.efi`

15. پارتیشن ESP در UEFI:

- ویژگی‌ها:
 - فرمت FAT32
 - پرچم `boot` و `esp`
 - حاوی فایل‌های efi.
- نحوه استفاده:
 - UEFI firmware ESP را شناسایی می‌کند
 - فایل efi. (مثل `shimx64.efi`) را اجرا می‌کند

16. تحلیل بخشی از grub.conf:

```
bash
} 'menuentry 'Ubuntu
# ثبت وضعیت بوت ناموفق      recordfail
# لود درایور ویدئو            load_video
# لود فشرده‌سازی gzip        insmod gzio
# پشتیبانی از پارتیشن GPT   insmod part_gpt
# پشتیبانی از فایل‌سیستم ext2 insmod ext2

# تنظیم پارتیشن روت 'set root='hd0,gpt2
# جستجوی پارتیشن براساس UUID ...search --fs-uuid --set=root 49ee8c4e

# لود کرنل linux /vmlinuz-5.4.0-65-generic root=/dev/mapper/vg0-root ro
# لود initramfs initrd /initrd.img-5.4.0-65-generic
{
```

پیاده‌سازی عملی پیشنهادی:

1. بررسی ساختار sys/ در سرور واقعی

2. تست دستورات lsblk, lspci

3. مشاهده دمای CPU از sysfs:

```
bash .4
```

```
"watch -n 1 "cat /sys/class/thermal/thermal_zone*/temp .5
```

6. شبیه‌سازی خرابی با حذف ماژول کرنل:

```
bash .7
```

```
sudo rmmod [module_name] && lsmod .8
```