

# LPIC1 (T2)

## Ghazaleh Keyvani

### 2025/7/7

XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX

1. مفهوم "Sector" در `fdisk`:

- سکتور: کوچکترین واحد آدرس‌دهی روی دیسک (معمولاً 512 یا 4096 بایت)
- در خروجی `fdisk -p`:
  - `Start`: شماره سکتور شروع پارتیشن
  - `End`: شماره سکتور پایان پارتیشن
  - `Sectors`: تعداد کل سکتورهای اختصاص‌یافته به پارتیشن
- محاسبه اندازه پارتیشن:

● `bash`

●  $(\text{End} - \text{Start} + 1) \times \text{اندازه\_سکتور}$

مثال:

```
Start=2048, End=1050623, Sector Size=512B  
(1050623 - 2048 + 1) × 512 = 537 MB
```

---

## 2. توصیه‌های پارتیشن‌بندی برای سرورها:

ملاحظات	اندازه‌های پیشنهادی	پارتیشن‌های ضروری	نوع سیستم
برای LVM استفاده از انعطاف‌پذیری	،(باقیمانده) /home (30GB) ، / swap (1.5×RAM)	، /home ، / swap ، /boot/efi	دسکتاپ لینوکس
جداسازی پارتیشن لاگ‌ها برای جلوگیری از پر شدن /	،(دیسک 20%) /var ،(دیسک 10%) /var/log ،(باقیمانده) /data	، /var ، /var/log ، /data ، swap	سرورهای دیتابیس/وب
user quotas فعال‌سازی برای محدودیت فضای کاربران	،(دیسک 50%) /tmp ،(دیسک 10%) /shared ،(دیسک 30%)	، /tmp ، /home ، /tmp ، /shared ، swap	سرورهای دانشگاهی

## 3. اجرای برنامه‌ها با حافظه ناکافی:

سیستم‌عامل از حافظه مجازی استفاده می‌کند:

1. صفحه‌بندی (Paging): تقسیم حافظه به بلوک‌های ثابت (صفحه‌ها)
2. تعویض صفحه (Swapping): انتقال صفحه‌های کم‌استفاده به فضای Swap روی دیسک
3. تکنیک‌های مدیریت:
  - Demand Paging: لود فقط صفحه‌های موردنیاز در RAM
  - Page Replacement: جایگزینی صفحه‌های قدیمی (الگوریتم‌هایی مثل LRU)
  - Thrashing: وقتی سیستم بیشتر زمان خود را صرف جابجایی صفحه‌ها کند (افت شدید عملکرد)

پاسخ به سوالات فرعی:

- نیاز به لود تمام صفحات؟ خیر، فقط صفحه‌های فعال کافی هستند
- وقتی RAM پر است: سیستم شروع به Swapping می‌کند
- استفاده از دیسک: فضای Swap به عنوان حافظه مجازی استفاده می‌شود

---

#### 4. TLB (Translation Lookaside Buffer):

- مفهوم: کش سخت‌افزاری برای نگاشت سریع آدرس‌های مجازی به فیزیکی
- نقش: کاهش تاخیر دسترسی به حافظه با جلوگیری از مراجعه به صفحه‌جدول
- مثال عملکرد:

1. CPU آدرس مجازی را تولید می‌کند
2. TLB چک می‌کند آیا ترجمه وجود دارد (TLB Hit)
3. اگر نبود (TLB Miss)، به صفحه‌جدول مراجعه می‌شود

---

#### 5. مفاهیم پایه:

- صفحه (Page): بلوک ثابت حافظه فیزیکی (معمولاً 4KB)
- صفحه مجازی (Virtual Page): بلوک حافظه در فضای آدرس‌دهی مجازی فرآیند
- تعویض زمینه (Context Switch):
  - عملیات ذخیره حالت فرآیند جاری و بارگذاری فرآیند جدید
  - شامل به‌روزرسانی رجیسترها، صفحه‌جدول و TLB Flush

تأثیرات:

#### فاکتور

#### تأثیر بر عملکرد

---

Swap افزایش فضای

(Swapping به دلیل) Context Switch افزایش تاخیر در

---

صفحه‌های بزرگ

TLB کاهش دفعات تعویض صفحه، افزایش کارایی

---

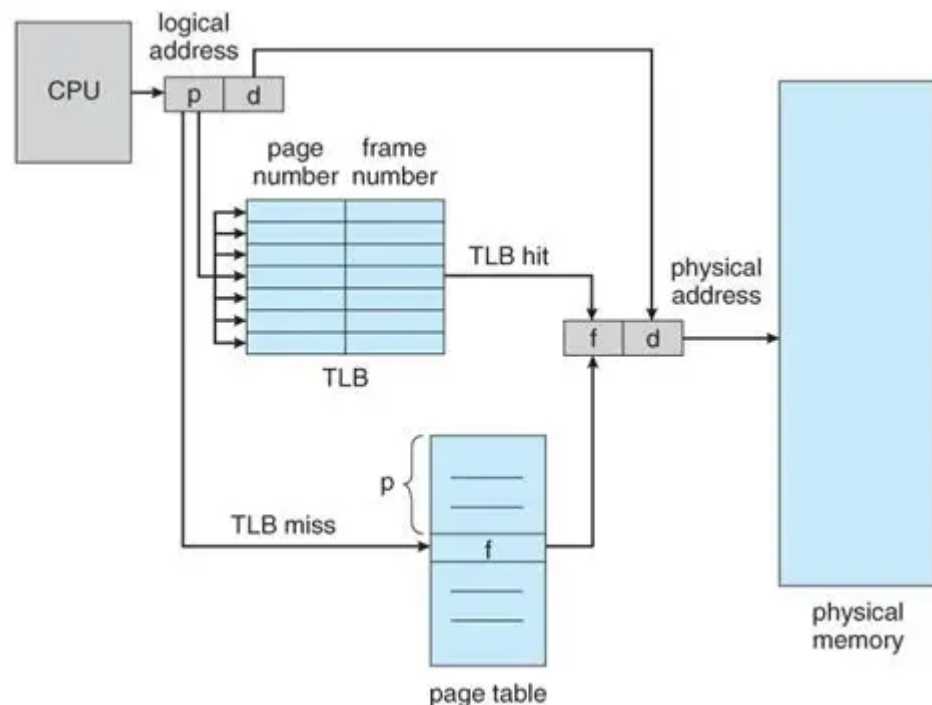
---

#### 6. Huge Pages:

- مفهوم: صفحه‌های حافظه بسیار بزرگتر از استاندارد (معمولاً 2MB یا 1GB)
- هدف:
  - کاهش فشار بر TLB
  - بهبود عملکرد برنامه‌های حافظه‌محور (دیتابیس‌ها، محاسبات علمی)
- استفاده:
  - bash

# فعال‌سازی در لینوکس

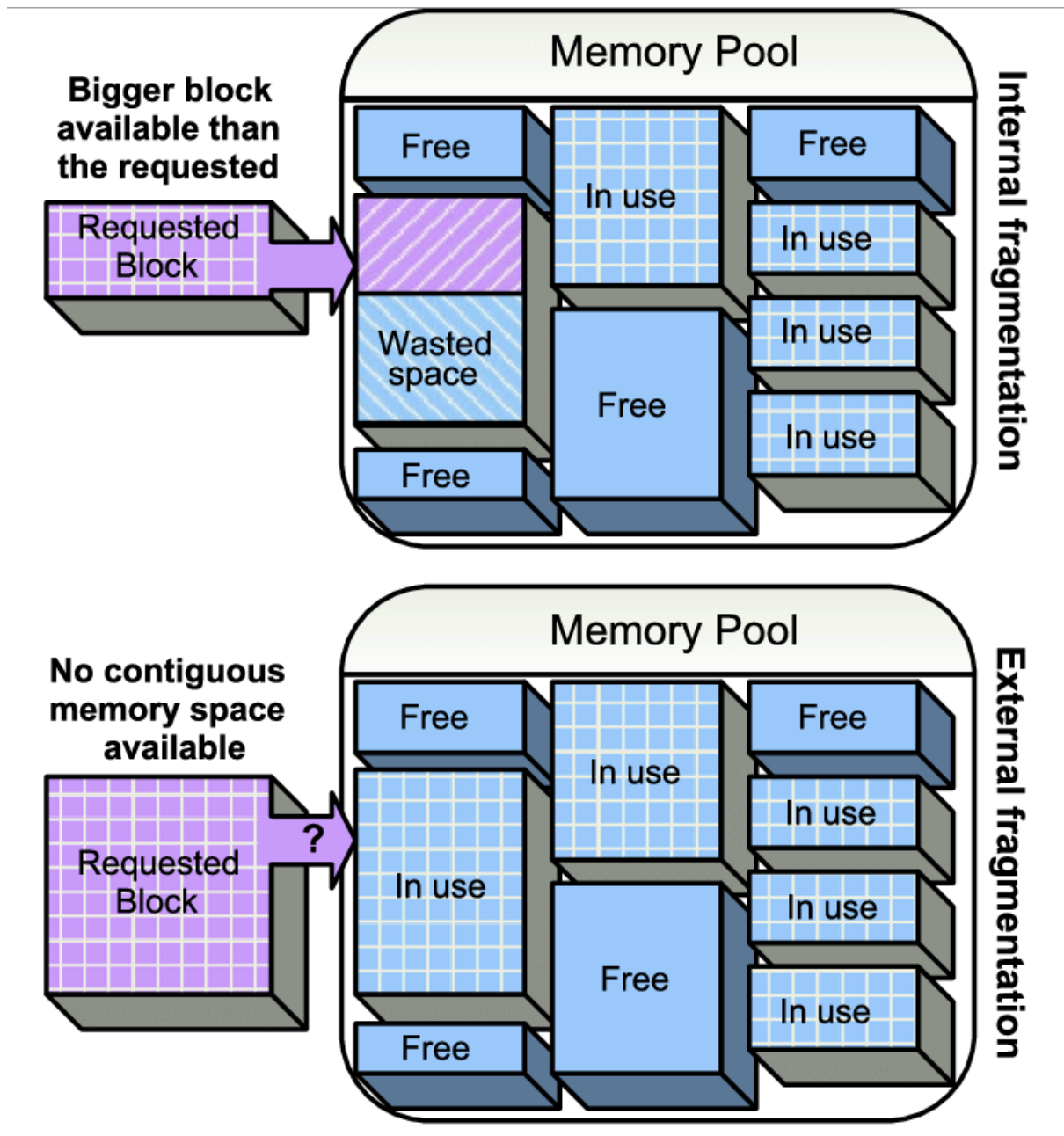
- `echo 2048 > /sys/kernel/mm/hugepages/hugepages-2048kB/nr_hugepages`
- مزایا: کاهش TLB Miss، کاهش سربار مدیریت حافظه
- معایب: اتلاف حافظه در برنامه‌های با نیاز کم



## 7. تکه‌تکه‌شدن حافظه (Fragmentation):

- انواع:
  - خارجی (External): فضاهای آزاد پراکنده و کوچک
  - داخلی (Internal): اتلاف فضای داخل بلوک‌های اختصاص‌یافته
- مشکلات:
  - عدم توانایی در تخصیص حافظه علیرغم وجود فضای کافی
  - افت عملکرد به دلیل جستجوی طولانی برای بلوک‌های آزاد

- افزایش مصرف حافظه
- راه‌حل‌ها:
  - Compaction: جابجایی بلوک‌ها برای یکپارچه‌سازی فضای آزاد
  - صفحه‌بندی: حل مشکل تکه‌تکه‌شدن خارجی
  - الگوریتم‌های تخصیص: Buddy System, Slab Allocation



1. مشاهده سکتورها:

2. bash

3. `sudo fdisk -l /dev/sda`

4. بررسی حافظه مجازی:

5. bash

# نمایش حافظه فیزیکی و Swap `free -h`

6. `vmstat 1` # مانیتورینگ وضعیت حافظه مجازی

7. مدیریت Huge Pages:

8. bash

# تنظیم تعداد Huge Pages

`echo 1024 | sudo tee /proc/sys/vm/nr_hugepages`

# مشاهده وضعیت

9. `cat /proc/meminfo | grep Huge`

10. شبیه‌سازی تکه‌تکه‌شدن:

11. c.

// برنامه C برای ایجاد تکه‌تکه‌شدن حافظه

`<include <stdlib.h>`

`int main`

`{ ++for(int i=0; i<1000; i`

`(void* ptr = malloc(rand() % 1000 + 1`

`{`

`;return 0`

`{`