

به نام خدا



دانشگاه صنعتی شریف

سیستم‌های عامل پیشرفته

نیم‌سال دوم ۰۱-۰۲

استاد: دکتر اسدی

دانشکده مهندسی کامپیوتر

---

تمرین سری سوم

- پاسخ تمرین‌های تئوری را به صورت فایل تایپ شده ( به زبان فارسی) در فرمت PDF در قسمت مربوطه در سامانه CW بارگذاری نمایید.
- پرسش‌های خود را می‌توانید در فروم ایجاد شده در سایت درس مطرح کنید.

1) برنامه ای به زبان C بنویسید که یک فایل به اندازه ۱۰ گیگابایت را ساخته و ۱۰ بار این فایل را بخواند. سپس برنامه ای دیگر بنویسید که با استفاده از پرچم O\_DIRECT<sup>۱</sup> حافظه‌ی نهان را bypass کرده و همان فایل را ۱۰ بار بخواند.

الف) زمان دقیق دو برنامه را در حالتی بدست آورده و با هم مقایسه کنید که قبل از اجرای برنامه دوم دستور زیر را وارد کرده باشید:

```
sync; echo 3 > /proc/sys/vm/drop_caches
```

ب) زمان دقیق دو برنامه را در حالتی بدست آورده و با هم مقایسه کنید که قبل از اجرای برنامه دوم دستور بالا را وارد نکرده باشید.

ج) دو برنامه گفته شده را اینبار برای نوشتن یک فایل ۱۰ گیگابایتی اجرا کرده و موارد الف و ب را انجام دهید. تاثیر Page Cache در خواندن بیشتر مشاهده میشود یا نوشتن؟

2) شما در حال کار بر روی پروژه ای هستید که شامل خواندن و نوشتن تعداد زیادی فایل کوچک است که هر کدام حاوی برخی فراداده<sup>۲</sup> خاص است. فراداده میتواند به سادگی یک شناسه یا یک نام باشد، اما باید به طور مداوم ذخیره شود و به سرعت قابل دسترسی باشد. وظیفه شما نوشتن برنامه ای است که بتواند این نیاز را به طور موثر برطرف کند. برای این منظور، یک برنامه C بنویسید که ۱۰۰۰ فایل کوچک ایجاد می کند که هر کدام شامل یک شناسه منحصر به فرد و یک نام تصادفی (۵ تا ۱۰ کاراکتر) است.

الف) زمان لازم برای نوشتن فراداده در هر فایل را بدون استفاده از حافظه نهان میانگیر<sup>۳</sup> اندازه گیری کنید.

ب) زمان لازم برای خواندن فراداده از هر فایل را بدون استفاده از حافظه نهان میانگیر اندازه گیری کنید.

ج) زمان لازم برای نوشتن فراداده در هر فایل را با استفاده از حافظه نهان میانگیر اندازه گیری کنید.

د) زمان لازم برای خواندن فراداده از هر فایل را با استفاده از حافظه نهان میانگیر اندازه گیری کنید.

و) نتایج ۴ حالت را مقایسه کرده و مزایای استفاده از حافظه نهان میانگیر را برای این سناریو توضیح دهید.

#### راهنما:

- از فراخوانی های سیستم open(), write(), read(), close() و lseek() برای مدیریت فایل I/O استفاده کنید.
- از تابع gettimeofday() از کتابخانه sys/time.h برای اندازه گیری زمان اجرا استفاده کنید.
- از پرچم O\_DIRECT با فراخوانی سیستم open() برای غیرفعال کردن حافظه‌ی نهان میانگیر استفاده کنید.
- از پرچم های O\_CREAT و O\_TRUNC با فراخوانی سیستم open() برای ایجاد فایل ها استفاده کنید.

<sup>1</sup> Flag

<sup>2</sup> Metadata

<sup>3</sup> Buffer Cache

- از پرچم‌های S\_IRUSR و S\_IWUSR با فراخوانی سیستم open() برای تنظیم مجوزهای فایل استفاده کنید.
- از lseek() برای انتقال نشانگر فایل به موقعیت صحیح قبل از نوشتن/خواندن متا داده استفاده کنید.
- از تابع rand() از کتابخانه stdlib.h برای تولید نام‌های تصادفی استفاده کنید.

3) در این بخش قصد داریم با چگونگی استفاده از یک ماژول نهان ساز ورودی/خروجی و تاثیر آن بر عملیات خواندن/نوشتن در حجم‌های بالا آشنا شویم. ابتدا ماژول/نرم افزار نهان ساز Open CAS را بر روی سیستم عامل لینوکس خود نصب کرده و تنظیمات اولیه آن را انجام دهید. پس از نصب حافظه‌ی نهان، از نسخه‌ی نصب شده عکس تصویر<sup>4</sup> تهیه کنید. در این ماژول، قسمتی از دیسک سخت (HDD) به عنوان دیسک پایه (Back-end) و قسمتی از دیسک حالت جامد (SSD) به عنوان حافظه نهان ورودی/خروجی در نظر گرفته میشود. همچنین، جهت انجام عملیات خواندن/نوشتن از دستور fio به عنوان یک ابزار محک ورودی/خروجی استفاده میکنیم. با دستور `sudo apt install fio` این پکیج را نصب نمایید.

\*\*\* در این تمرین ترجیحاً نیاز به داشتن هر دو دیسک سخت و دیسک حالت جامد می‌باشد. اما در صورت عدم در اختیار داشتن دیسک حالت جامد، میتوانید قسمتی از دیسک سخت را به عنوان دیسک سریع در نظر گرفته و برای نهان سازی به Open CAS معرفی نمایید.

الف) ابتدا با دستور زیر، یک فایل ۲۰ گیگابایتی بر روی یک پارتیشن از دیسک سخت خود (HDD) بنویسید و زمان اجرای آن را به صورت دقیق گزارش کنید:

```
fio --filename=/mnt/hdd_path/file.txt --direct=1 --rw=write --bs=4M --ioengine=libaio --iodepth=8 --numjobs=1 --group_reporting --io_size=20G --size=20G --name=iops-test-job --eta-newline=1 --output=seq-write-20.txt
```

ب) سپس با دستور زیر عملیات خواندن ترتیبی را بر دیسک سخت شبیه‌سازی کنید و زمان اجرا، میزان پهنای باند دیسک (به ازای خواندن و نوشتن) و تعداد عملیات ورودی/خروجی در واحد زمان (IOPS) و ... را ثبت نمایید. (عکس تصویر از خروجی fio پس از اتمام اجرا گرفته شود)

```
fio --filename=/mnt/hdd_path/file.txt --random_distribution=zipf:1.2 --direct=1 --rw=read --bs=4M --ioengine=libaio --iodepth=8 --numjobs=1 --group_reporting --io_size=20G --size=20G --name=iops-test-job --eta-newline=1 --output=seq-read-20.txt
```

---

<sup>4</sup> Screenshot

ج) فضای ۵ گیگابایت از SSD (یا در صورت عدم وجود، قسمتی از دیسک سخت) خود را به عنوان حافظه‌ی نهان تعریف نمایید و تمامی فضای دیسک سخت که در قسمت ب تعریف کرده بودید را به عنوان Backend حافظه‌ی نهان مشخص نمایید. پیکربندی کامل حافظه‌ی نهان خود را در گزارش آورده و توضیح دهید. در صورتی که حافظه‌ی نهان با موفقیت ایجاد شده باشد، توسط دستور lsblk مسیر /dev/cas1\_1 مشاهده می‌شود. دستور زیر را در ترمینال وارد نموده و عکس تصویر آن را تهیه کنید:

#### casadm -L

د) دیوایس /dev/cas1\_1 را به مسیری که دیسک سخت را نصب<sup>۵</sup> کرده بودید (در مثال تمرین /mnt/hdd\_path/file.txt نصب نمایید. سپس دستور قسمت ب را این بار با حافظه‌ی نهان نصب شده با سه سیاست مختلف Write Through، Write Back و Write-Around اجرا نمایید و زمان دقیق آن را گزارش کنید. تفاوت یا شباهت‌های میان اجرا با این سه سیاست را شرح دهید. همچنین نتایج اجرای این دستور را با حالتی که از حافظه‌ی نهان استفاده نمی‌شد (قسمت ب) مقایسه کنید و دلیل شباهت و تفاوت مشاهده شده با قسمت ب را توجیه کنید. در تمامی دستورات fio، زمان اجرا، میزان پهنای باند دیسک (به ازای خواندن و نوشتن) و تعداد عملیات ورودی/خروجی در واحد زمان (IOPS) و ... را ثبت نمایید. (عکس تصویر از خروجی fio پس از اتمام اجرا گرفته شود)

ح) مراحل (ب تا د) را برای دستور زیر نیز انجام دهید و مشابه قسمت د، گزارش آن را تهیه کنید:

```
fio --filename=/mnt/hdd_path/file.txt --random_distribution=zipf:1.2 --direct=1 --rw=randread --bs=4k --  
ioengine=libaio --iodepth=16 --numjobs=16 --group_reporting --size=20G --name=iops-test-job --eta-newline=1 --  
output=random-read.txt
```

#### توجه:

- پیشنهاد میشود دستورات را با استفاده از screen در لینوکس اجرا نمایید تا در صورت قطع برق، اجرا ادامه پیدا کند.
- زمانی که fio جهت اجرا تخمین می‌زد در آغاز اجرا ممکن است دقیق نباشد.
- جهت گزارش زمان دقیق اجرا می‌توانید از دستور time در لینوکس استفاده نمایید.
- پیش از هر اجرا، دستور sync; echo 3 > /proc/sys/vm/drop\_caches را در ترمینال وارد کنید.

#### • منابع مفید برای سوال ۳:

- 1) <https://open-cas.github.io>
- 2) <https://fio.readthedocs.io/en/latest/>

---

<sup>5</sup> Mount

### نکات مهم تمرین:

- پاسخ سوالات به زبان فارسی نوشته شوند.
- تمامی برنامه ها بایستی به زبان C نوشته شوند.
- نکات ذیل میبایست در عکس‌های گرفته شده رعایت شود:
  - تصاویر واضح و شفاف باشند.
  - زمان و تاریخ کنونی سیستم (شمسی یا میلادی) مشخص باشد.
  - نام کاربری در محیط command line شامل نام و نام خانوادگی و شماره دانشجویی فرد به فرمت زیر باشد:

**firstName\_lastName\_studentID**

- خروجی این تمرین، یک فایل PDF شامل تمام تصاویر و توضیحات هر سوال، به همراه کدهای نوشته شده می‌باشد.
- تمامی کدهای نوشته شده (شامل برنامه‌های به زبان C) توسط تیم دستیاران بررسی و اجرا می‌شوند. بنابراین، حتما قبل از ارسال از کارکرد صحیح کدها و اسکرپت‌ها اطمینان حاصل نمایید.
- استفاده از کدهایی که ممکن است در اینترنت بیابید، مجاز نیست و شباهت کدهای شما با کدهای آماده، یا کدهای سایر دانشجویهای درس، به منزله‌ی تقلب و ثبت نمره‌ی **صفر** خواهد بود.

**موفق باشید**