نام درس: شبکه های کامپیوتری مدرس: دکتر حسینی سنو مهر ۱۴۰۵-۱۴۰۴	گروه مهندسی کامپیوتر دانشگاه فردوسی مشهد	مهلت تحویل: ۱۴/ ۹ / ۱۴۰۴

تمرین عملی

شرح

پروژه‌ی WebCacheX یک شبیه‌ساز آموزشی سه‌لایه در حوزه‌ی شبکه‌های کامپیوتری است که با استفاده از سوکت‌نویسی طراحی می‌شود.

این سامانه شامل سه مؤلفه‌ی اصلی است:

۱. مرورگر کلاینت (Browser Client)
۲. کش‌سرور میانی (Cache Server)
۳. وب‌سرور اصلی (Web Server)

هدف از این پروژه، درک عمیق‌تر نحوه عملکرد پروتکل HTTP ، مفاهیم Caching ، مدیریت درخواست‌ها، و بهینه‌سازی ترافیک شبکه در معماری‌های چندنقطه‌ای است.

در این سیستم، مرورگر به‌جای ارتباط مستقیم با وب‌سرور، درخواست‌های خود را به کش‌سرور ارسال می‌کند.

کش‌سرور در صورت وجود پاسخ معتبر در حافظه، آن را مستقیماً به مرورگر بازمی‌گرداند؛ در غیر این‌صورت، درخواست را به وب‌سرور اصلی ارسال کرده، پاسخ را دریافت و ذخیره کرده و سپس برای کلاینت ارسال می‌کند.

این فرآیند شبیه‌سازی دقیقی از نحوه عملکرد Proxy Server ها و CDN Cache ها در وب واقعی است.

مؤلفه‌های اصلی پروژه

۱. وب‌سرور اصلی(Web Server)

- وظیفه پاسخ‌گویی به درخواست‌های HTTP را برعهده دارد.
- محتوای درخواستی (HTML ، JSON، عکس یا متن) را از مسیر مشخصی مثلاً (/web_content) می‌خواند.
- پاسخ باید با ساختار استاندارد HTTP ارسال شود، شامل هدرهایی مانند:
- HTTP/۱,۱ ۲۰۰ OK
- Content-Type: text/html
- Cache-Control: max-age=۱۰
- در صورتی که فایل مورد نظر وجود نداشته باشد، باید پاسخ ۴۰۴ Not Found برگردانده شود.
- هر پاسخ باید دارای مدت اعتبار کش (TTL) باشد تا کش‌سرور بتواند زمان انقضای محتوا را محاسبه کند.

۲. کش‌سرور میانی(Cache Server)

- نقش وساط میانی بین مرورگر و وب‌سرور را ایفا می‌کند.
- در زمان دریافت درخواست از مرورگر، مراحل زیر را انجام می‌دهد:
  ۱. بررسی می‌کند آیا صفحه در **حافظه کش (Cache)** موجود است.
  ۲. اگر موجود باشد و زمان آن منقضی نشده باشد، مستقیماً به مرورگر پاسخ می‌دهد.**(Cache HIT)**
  ۳. اگر در کش وجود نداشته باشد یا منقضی شده باشد **(Cache MISS)** ، درخواست را به وب‌سرور اصلی ارسال می‌کند.
  ٤. پاسخ دریافتی را در حافظه ذخیره کرده و سپس برای مرورگر ارسال می‌کند.
  ٥. تمامی رویدادها (HIT، MISS، Expire) باید در فایل لاگ ثبت شوند.
- کش‌سرور باید:
  - چنددرخواستی باشد**(Multi-threaded)**
  - داده‌ها را با زمان انقضای مشخص نگهداری کند (مثلاً max-age=۱۰s)
  - ورودی‌های منقضی‌شده را حذف نماید**(Cache Eviction)**
  - گزارش لحظه‌ای عملکرد خود را در فایل cache_log.txt ذخیره کند.

۳. مرورگر کلاینت(Browser Client)

- نقش کاربر نهایی را شبیه‌سازی می‌کند.
- به‌جای اتصال مستقیم به وب‌سرور، درخواست‌های خود را به کش‌سرور می‌فرستد.
- خروجی دریافتی از کش‌سرور را در صفحه یا ترمینال نمایش می‌دهد.
- درخواست‌ها در قالب ساده HTTP ارسال می‌شون

نحوه عملکرد سیستم

۱. مرورگر درخواستی برای فایل خاص مثلاً (index.html)به کش‌سرور ارسال می‌کند.
۲. کش‌سرور بررسی می‌کند آیا فایل در حافظه کش موجود است:
  - اگر بله و هنوز معتبر است، پاسخ را مستقیماً به مرورگر می‌دهد. **(HIT)**
  - اگر خیر، درخواست را برای وب‌سرور اصلی ارسال می‌کند. **(MISS)**
۳. وب‌سرور فایل را پیدا کرده و همراه با زمان انقضا پاسخ می‌دهد.
٤. کش‌سرور پاسخ را در حافظه ذخیره و به مرورگر بازمی‌گرداند.
٥. در درخواست‌های بعدی، همان پاسخ بدون نیاز به تماس با وب‌سرور داده می‌شود تا تأخیر کاهش یابد.

ساختار گزارش(Logging)

هر درخواست باید با جزئیات در فایل لاگ ذخیره شود، شامل نوع نتیجه و زمان پاسخ‌گویی.

```
[2025-10-04 14:22:03] Request: /index.html | Result: HIT | Served from Cache
[2025-10-04 14:22:15] Request: /page2.html | Result: MISS | Fetched from WebServer
[2025-10-04 14:22:25] Cache Expired: /index.html | Revalidated
```

الزامات فنی

۱. استفاده از هر زبان برنامه‌نویسی برای پیاده‌سازی پروژه **بلامانع** است.
۲. نوع ارتباط بین مؤلفه‌ها باید **مبتنی بر پروتکل TCP** باشد.
۳. تمامی تبادلات داده باید بر اساس **پروتکل HTTP ساده‌شده** انجام شوند.
٤. ساختار کلی سیستم باید به‌صورت **سه‌لایه**(**Client** ، **Cache Server**، **Web Server**) طراحی شود مؤلفه‌ی مرورگر (Client) باید دارای **رابط کاربری گرافیکی (GUI)** جهت ارسال درخواست‌ها و نمایش پاسخ‌ها باشد.
٥. سرورها باید از **قابلیت چندکاربره (Multithreading)** پشتیبانی کنند تا امکان دریافت هم‌زمان چند درخواست وجود داشته باشد.
٦. تمامی رویدادها (از جمله Cache HIT ، Cache MISS، Expire، Error) باید در فایل گزارش **cache_log.txt**ثبت شوند.
۷. زمان اعتبار محتوای ذخیره‌شده در کش (**TTL**) باید برابر با **۱۰ ثانیه** باشد.
۸. در صورت عدم وجود فایل در وب‌سرور، باید پاسخ **HTTP ٤٠٤ Not Found**بازگردانده شود.
۹. پروژه باید **به‌صورت تک‌نفره** انجام شود.
۱۰. کد منبع باید دارای ساختار منظم و ماژولار بوده و با **توضیحات (Docstring)** و **کامنت‌های کافی** مستند شده باشد.
۱۱. مستندات پروژه باید شامل موارد زیر باشند:
۱۲. فایل **README.md**با توضیح مراحل اجرا و نمونه خروجی
۱۳. فایل **cache_log.txt**یا **report.log**برای گزارش عملکرد سیستم
۱٤. توضیح کوتاه از تست‌های انجام‌شده جهت اثبات صحت عملکرد کش
۱٥. ارزیابی پروژه به‌صورت **دستی (Code Review)** انجام می‌شود و معیار اصلی، صحت عملکرد فنی و رعایت اصول طراحی سه‌لایه است.

نمره اضافه

۱. پشتیبانی از چند وب‌سرور(Multi-Origin Support) : کش‌سرور بتواند چند وب‌سرور مختلف را پشتیبانی کند و بر اساس دامنه یا پورت، به سرور مناسب درخواست بفرستد.

هر مورد خلاقانه دیگری در صورتی هماهنگی با تیم حل تمرین می تواند نمره اضافه لحاظ شود