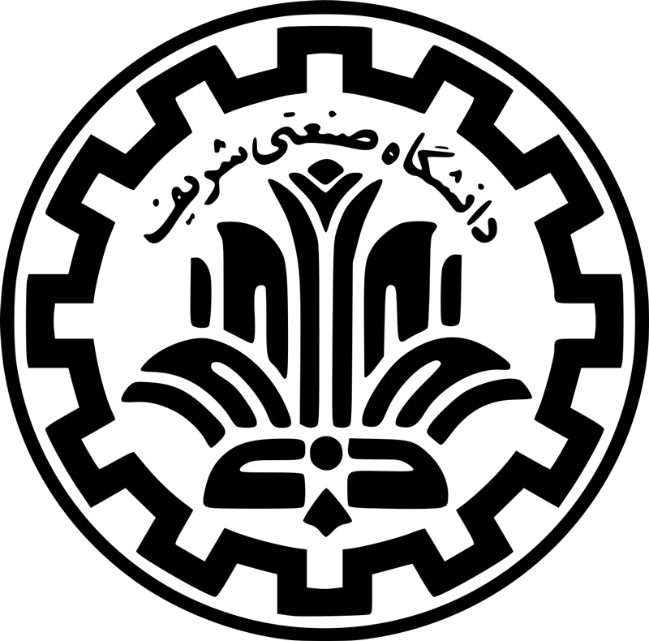
**به نام خدا**



**سیستم های نهفته بی درنگ**

تمرین شماره 4

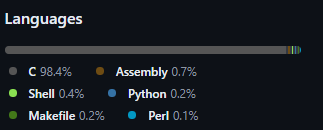
**استاد: دکتر غلامپور**

**دانشجویان:** حسین انجیدنی 400100746

**تیر ماه** 1403

**سوال اول**

**الف)**

****

تصویر توزیع زبان ها به کار رفته در کرنل لیتوکس

سیستم عامل لینوکس عمدتاً به زبان C نوشته شده است، اما برخی از قسمت‌های آن به زبان اسمبلی نوشته شده‌اند. این بخش‌ها شامل قسمت‌هایی هستند که نیاز به دسترسی مستقیم به سخت‌افزار دارند یا باید با کارایی بسیار بالایی اجرا شوند. در زیر به تفصیل به این بخش‌ها و دلایل استفاده از زبان اسمبلی در هر کدام پرداخته شده است:

**1. راه‌اندازی (Bootstrap)**

* **بخش اسمبلی:** کدهای اولیه راه‌اندازی (bootloader) و کدهای ابتدایی کرنل که سیستم را از حالت بوت به حالت عملیاتی می‌برند.
* **علت استفاده:** در این مرحله، سیستم هنوز به طور کامل راه‌اندازی نشده و محیط اجرایی لازم برای اجرای کدهای C فراهم نیست. بنابراین، از زبان اسمبلی استفاده می‌شود که مستقیماً با سخت‌افزار در ارتباط است و می‌تواند وظایف اولیه مثل تنظیمات اولیه پردازنده و حافظه را انجام دهد.

**2. مدیریت وقفه‌ها (Interrupt Handling)**

* **بخش اسمبلی:** کدهای مدیریت وقفه‌ها و استثناها.
* **علت استفاده:** این کدها باید با سرعت و کارایی بالا اجرا شوند و مستقیماً با سخت‌افزار پردازنده در ارتباط باشند. زبان اسمبلی به دلیل سرعت بالا و توانایی دسترسی مستقیم به رجیسترها و منابع پردازنده، برای این منظور مناسب است.

**3. مدیریت حالت هسته و کاربر (Kernel and User Mode Switching)**

* **بخش اسمبلی:** کدهای سوئیچ کردن بین حالت هسته و حالت کاربر.
* **علت استفاده:** این کدها باید با کارایی بسیار بالا و به صورت بهینه اجرا شوند. همچنین، نیاز به دسترسی مستقیم به رجیسترها و تنظیمات پردازنده دارند که با زبان اسمبلی به راحتی امکان‌پذیر است.

**4. روال‌های خاص پردازنده (Processor-Specific Routines)**

* **بخش اسمبلی:** توابع و روال‌هایی که به صورت خاص برای پردازنده‌های مختلف نوشته شده‌اند.
* **علت استفاده:** این توابع برای بهره‌برداری کامل از قابلیت‌ها و ویژگی‌های خاص هر پردازنده نوشته شده‌اند و نیاز به دسترسی مستقیم به رجیسترها و تنظیمات خاص دارند.

**5. بهینه‌سازی‌های سطح پایین (Low-Level Optimizations)**

* **بخش اسمبلی:** بخش‌هایی از کد که نیاز به بهینه‌سازی‌های بسیار خاص و دقیق دارند.
* **علت استفاده:** زبان اسمبلی به برنامه‌نویس اجازه می‌دهد تا کنترل کامل بر روی کد و سخت‌افزار داشته باشد و از این طریق می‌تواند بهینه‌سازی‌های دقیقی را اعمال کند که با زبان C امکان‌پذیر نیست.

**نتیجه‌گیری**

استفاده از زبان اسمبلی در کرنل لینوکس به دلایل کارایی، دسترسی مستقیم به سخت‌افزار، و نیاز به اجرای بهینه و سریع در برخی بخش‌های خاص ضروری است. هر چند که بیشتر کرنل به زبان C نوشته شده است تا خوانایی و نگه‌داری کد آسان‌تر شود، اما در بخش‌هایی که نیاز به دسترسی مستقیم به سخت‌افزار و کارایی بالا وجود دارد، از زبان اسمبلی استفاده می‌شود.

**ب)**

**آسیب‌پذیری‌های بحرانی اخیر در هسته لینوکس**

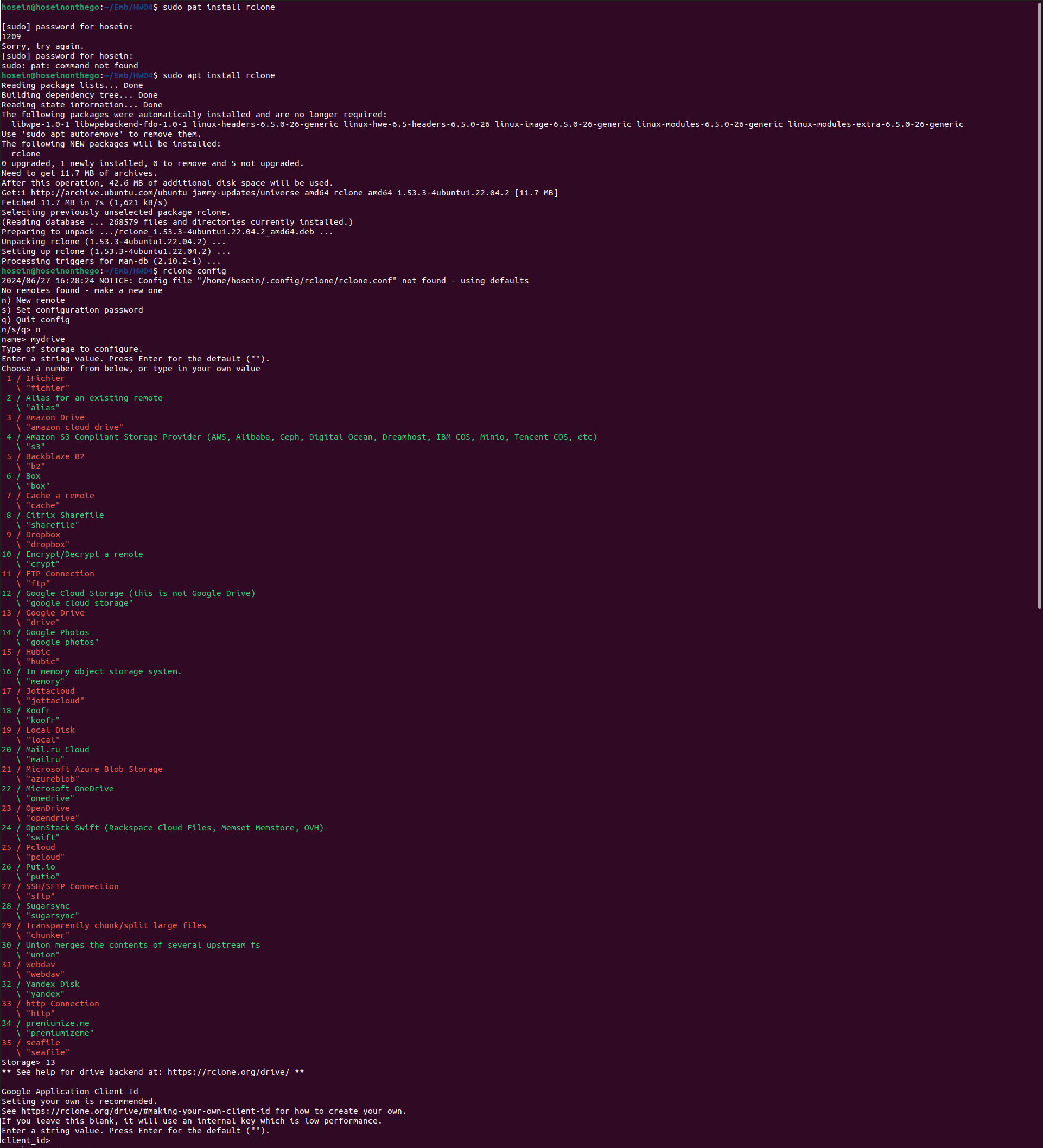
در اینجا به پنج آسیب‌پذیری بحرانی اخیر در هسته لینوکس اشاره می‌کنیم:

1. **آسیب‌پذیری اجرای کد از راه دور (CVE-2022-47939)**
   * **شرح:** این آسیب‌پذیری استفاده پس از آزادسازی (use-after-free) در ماژول سرور فایل SMB به نام ksmbd در هسته لینوکس می‌تواند به مهاجم غیرمجاز از راه دور اجازه دهد که کد دلخواه را اجرا کند. مشکل در تابع DISCONNECT\_TREE\_2SMB است که قبل از انجام عملیات بر روی شیء، وجود آن را بررسی نمی‌کند.
   * **نمره CVSSv3:** 10.0 (بحرانی)
   * **زمان تشخیص و رفع:** این باگ در تاریخ 17 اوت 2022 در نسخه 5.15.61 هسته لینوکس رفع شد. رفع این باگ شامل تصحیح مدیریت حافظه در تابع DISCONNECT\_TREE\_2SMB بود. کاربران سیستم‌های آسیب‌پذیر باید پچ‌های موجود را اعمال کنند.
   * **منبع:** [Tenable](https://www.tenable.com/blog/cve-2022-47939-critical-rce-vulnerability-in-linux-kernel)
2. **آسیب‌پذیری Dirty Pipe**
   * **شرح:** این آسیب‌پذیری که توسط مکس کلرمن کشف شد، بر نسخه‌های 5.8 و بعد از هسته لینوکس تأثیر می‌گذارد. این باگ به مهاجم اجازه می‌دهد تا داده‌های فایل‌های فقط خواندنی را بازنویسی کند و منجر به افزایش دسترسی شود.
   * **زمان تشخیص و رفع:** این نقص در به‌روزرسانی‌های اخیر هسته لینوکس رفع شده است. مشکل با تنظیم فلگ‌های بافر پایپ در کد منبع هسته لینوکس رفع شد.
   * **منبع:** TechRadar
3. **تقسیم بر صفر در درایور نمایش AMD**
   * **شرح:** یک باگ در درایور نمایش AMD مربوط به تنظیم پیکربندی فشرده‌سازی جریانی (DSC) می‌تواند منجر به خطای تقسیم بر صفر شود وقتی پارامتر height\_slice برابر صفر تنظیم شود. این باگ می‌تواند کرنل را خراب کند و نیاز به راه‌اندازی مجدد سیستم دارد.
   * **زمان تشخیص و رفع:** این باگ در آخرین به‌روزرسانی‌های هسته لینوکس رفع شده است. رفع این باگ شامل تصحیح نحوه مدیریت پارامترهای پیکربندی DSC در درایور نمایش AMD بود.
   * **منبع:** [The Register](https://www.theregister.com/2024/03/29/linux_kernel_flaw/)
4. **افزایش دسترسی در نسخه‌های کرنل 5.14 تا 6.6.14**
   * **شرح:** یک آسیب‌پذیری جدید کشف شده به یک مهاجم محلی اجازه می‌دهد تا به دسترسی ریشه (root) در سیستم آسیب‌پذیر دست یابد با استفاده از یک نقص در مدیریت برخی از فراخوانی‌های سیستمی.
   * **زمان تشخیص و رفع:** این آسیب‌پذیری در به‌روزرسانی‌های اخیر هسته لینوکس رفع شده است. راه‌حل شامل تصحیح مدیریت فراخوانی‌های سیستمی آسیب‌پذیر بود.
   * **منبع:** [The Register](https://www.theregister.com/2024/03/29/linux_kernel_flaw/)
5. **بایپس احراز هویت در ماژول SFTP انتقال MOVEit**
   * **شرح:** این آسیب‌پذیری بر ماژول SFTP انتقال MOVEit تأثیر می‌گذارد و می‌تواند تحت شرایط خاصی منجر به بایپس احراز هویت شود.
   * **زمان تشخیص و رفع:** این آسیب‌پذیری به تازگی کشف و رفع شده است. رفع این مشکل شامل تصحیح احراز هویت در ماژول SFTP بود.
   * **منبع:** [The Register](https://www.theregister.com/2024/03/29/linux_kernel_flaw/)

توصیه می‌شود که کاربران سیستم‌های آسیب‌پذیر تمامی به‌روزرسانی‌ها و پچ‌های امنیتی موجود را اعمال کنند تا از سوءاستفاده‌های احتمالی جلوگیری شود.

**ج)** تمام فرایند ها توسط دستورات لینوکس و برنامه rclone انجام شده که در تصویر زیر قابل مشاهده میباشد.

فایل در این لینک قابل دسترسی میباشد (لازم به ذکر است که این لینک نیز توسط rclone ساخته شده.)



**سوال دوم**

خروجی های این سوال در یک فیلم ده دقیقه ای نمایش داده شده است.

در این سوال از libssh استفاده شده است و تمامی خروجی ها و مراحل کار با آن در فیلم آمده است.

فرایند اجرای cmake به شکل زیر است:

Cmake -DRH=”server adddress” -DUN=”username” -DPASS=”password” -DCPU=cpu\_limit -DMEM=memory\_limit

در فایل های SSH\_Connection توابعی موجود است که به ssh متصل میشود و توسط تابع execute دستور مورد نظر ارسال میشود.

فایل CMAKE List:

در این فایل افزون بر معرفی فایل ها برای ساخت فایل make متغیر های موجود در دستور قبل تعریف شده است که جایگزین در فایل main.cpp میشود.

**سوال سوم**

**الف)**

این کد با استفاده از هدر های فایرفاکس نوشته شده است؛ همچنین خروجی های این کد به شرح زیر است که در تصویر صفحه بعد آمده است.

**ب)**

خروجی این بخش در فایل news.txt موجود است. که به شرح زیر است:

Scraped on 2024-06-29 15:13:47

عضویت استاد دانشگاه صنعتی شریف در هیئت تحریریه نشریه معتبر از انجمن شیمی آمریکا (ACS)

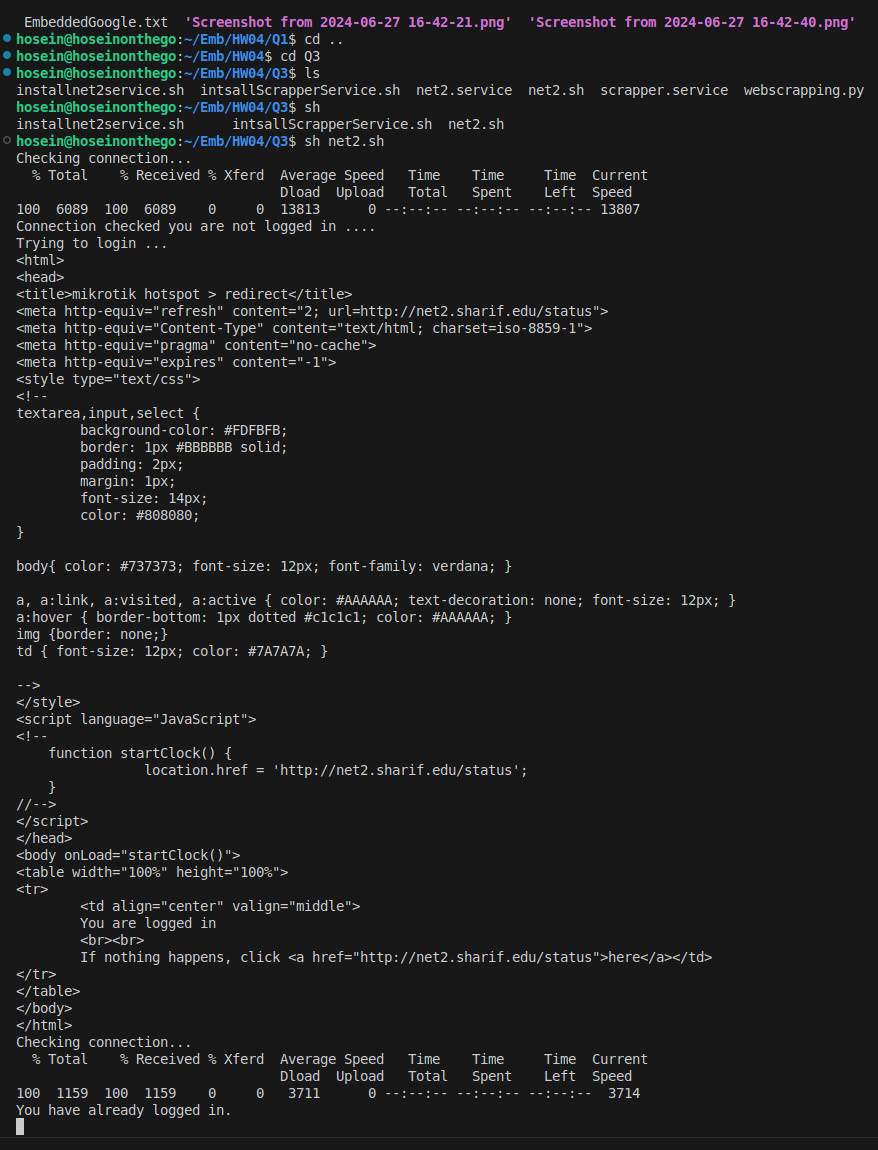
دانشگاه صنعتی شریف میزبان میهمانان جشن بزرگ غدیر

آغاز عملیات اجرایی ساخت استادسرای پونک دانشگاه صنعتی شریف

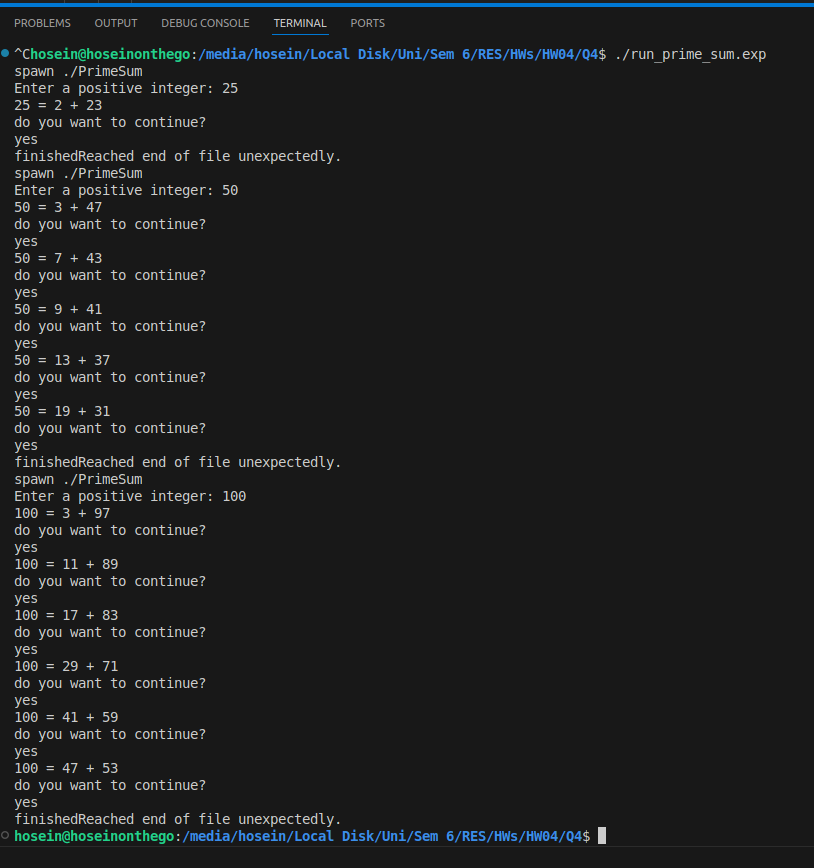
نصب نشان عالی دانش بر سینه استاد برجسته دانشگاه صنعتی شریف

آخرین احکام صادره

برگزاری مراسم دعای عرفه در مسجد دانشگاه



**سوال چهارم:**

****

**همچنین خروجی GDB در فایل دیباگ موجود است.**