باسمه تعالی

دانشگاه صنعتی شریف

دانشکده مهندسی برق

تمرین سری چهارم

دستیاران آموزشی : محمد جواد محمدی، علی قنبریان

استاد درس : دکتر ایمان غلامپور

نیم‌سال دوم ۲-۱۴۰۳

۱ تمرین اول: آشنایی اولیه با سیستم عامل Linux

در این تمرین قرار است کمی با بعضی از بخش‌های سیستم عامل لینوکس آشنایی پیدا کنید. به سوالاتی که در ادامه مطرح می‌شود تا جایی که ممکن است دقیق و کامل و در عین حال مختصر توضیح دهید. پر واضح است که هر چه (خصوصا بخش‌های تحقیقاتی) دقیق‌تر و کامل‌تر انجام دهید ارزش بیشتری دارد و شایسته نمره بالاتری است.

(الف) همانطورکه در کلاس درس هم اشاره شده است، اکثر کد کرنل لینوکس به زبان C نوشته شده است ولی برخی بخش‌های آن به زبان اسمبلی است. تحقیق کنید که اولا دقیقا آن بخش‌هایی که به زبان اسمبلی نوشته شده‌اند چه هستند و ثانیا اینکه علت هر یک از آنها به چه دلیلی است. در این باره تحقیق کنید.

(ب) تحقیق کنید که اخیرا چه اشکالات یا باک‌هایی در سیستم‌عامل لینوکس پیدا شده است. به حداقل ۵ مورد از آنها با ذکر منبع (یا کد مربوطه) اشاره کنید و دقیق توضیح بدهید و زمان تشخیص و رفع آنها را نیز بنویسید.

(ج) در این بخش می‌خواهیم با کاربرد ساده ولی بسیار مهمی آشنا شویم. با استفاده از دستورات لینوکس، باید این کار را انجام بدهید. ابتدا یک فایل text خالی را با نام EmbeddedGoogle در گوگل درایو خود قرار بدهید. سپس آن را دانلود کنید، سپس زمان دقیق آن لحظه را با دستورات مربوطه به فرمت مناسب (تاریخ شامل روز، ماه، سال و همچنین زمان شامل ساعت، دقیقه و ثانیه) را وارد کنید به همراه نام و نام خانوادگی و شماره دانشجویی‌تان و شماره‌ای که با آن عضو کانال رسمی درس هستید به همراه آدرس جیمیل‌تان که مرتبط با گوگل درایوتان است (در خطوط جداگانه) سپس آن را مجدد در گوگل درایو خود آپلود نمائید. (آن را Open Access کنید و لینک آن را در گزارش بگذارید.) در گزارش تان حتما تصاویر کافی به همراه توضیحات از تمام مراحل کار باشد.

۲ تمرین دوم: Make - CMake

در این تمرین قرار است که برنامه یک remote monitor را بنویسید و به درستی آن را اجرا کنید. کد این برنامه را به زمان C++ می‌نویسید و برای بهره‌برداری از آن از Make و CMake استفاده می‌نمائید.

برای انجام این تمرین در بخش برنامه کد C++ باید از پروتکل ارتباطی SSH استفاده کنید به گونه‌ای که از یک سیستم دارای سیستم عامل لینوکس به یک سیستم دیگر متصل شوید و پس از تبادل یک سری پیغام‌های اولیه که نشان دهنده صحت ارتباط بین این دو است و یک سری اطلاعات مهم درباره مشخصات پردازنده، حافظه سیستم ثانویه و پروتکل SSH واسط بین دو سیستم را در سیستم اولیه نشان می‌دهد، اگر هر یک از میزان‌های (درصدهای) استفاده از پردازنده و حافظه در سیستم دوم از آستانه‌های مشخصی که توسط کاربر در سیستم اول تعیین می‌شوند فراتر روند، پس از نشان دادن اطلاعات سخت‌افزاری میزان استفاده، هشداری درباره آن در سیستم اول به نمایش در آید.

بلافاصله آن process ای که بیشترین سهم را در این اتفاق داشته است پیدا کند و آی دی آن را نشان بدهد و سپس آن را متوقف کند.

این کار باید هر ۳۰ ثانیه یک بار اتفاق بیفتد یعنی باید به گونه‌ای برنامه را بنویسید که به صورت اتوماتیک هر ۳۰ ثانیه یک بار سیستم اولیه از طریق SSH سخت‌افزار سیستم ثاونیه را مانیتور کند و پیغام‌ها و عملیات‌های لازم که در بالا شرح داده شد را انحام بدهد.

سپس باید یک Makefile و CMakeLists استاندارد بنویسید که پروژه را build کند.

در نهایت باید پروژه شما چنین شکل و شمایلی داشته باشد.

```
remote_monitor/
├── src/
│   ├── main.cpp
│   ├── ssh_connection.cpp
│   └── hardware_monitor.cpp
├── include/
│   ├── ssh_connection.h
│   └── hardware_monitor.h
├── CMakeLists.txt
├── Makefile
└── README.md
```

البته پر واضح است که این نمایشی که ملاحظه کردید پیش از build شدن پروژه توسط Makefile و CMakeLists است و شما در فایلی که می‌فرستید علاوه‌بر فایل توضیحات، در این پوشه باید پوشه‌هایی که حاصل از build شدن پروژه ایجاد می‌شوند نیز موجود باشد. در نتیجه باید در کدهای Makefile و CMakeLists به این نکته توجه کنید که اگر آن پوشه‌ها یا فایل‌های ناشی از build شدن پروژه از قبل وجود دارند باید ابتدا پاک شوند و سپس از نو ایجاد شوند.

به عنوان نمونه، خروجی نهایی پروژه می‌تواند به شکل زیر باشد:

```
SSH connection established and authenticated successfully!
Connecting to remote host...
Connection established.


System CPU Information:
Architecture: x86_64
CPU op-mode(s): 32-bit, 64-bit
Byte Order: Little Endian
CPU(s): 4
On-line CPU(s) list: 0-3
Thread(s) per core: 2
Core(s) per socket: 2
Socket(s): 1
NUMA node(s): 1
Vendor ID: GenuineIntel
```

```
CPU family: 6
Model: 158
Model name: Intel(R) Core(TM) i5-8250U CPU @ 1.60GHz
Stepping: 10
CPU MHz: 800.078
CPU max MHz: 3400.0000
CPU min MHz: 400.0000
BogoMIPS: 3600.00
Virtualization: VT-x
L1d cache: 32K
L1i cache: 32K
L2 cache: 256K
L3 cache: 6144K
NUMA node0 CPU(s): 0-3

 System Memory Information:
total used free shared buff/cache available
Mem: 7.7G 2.3G 3.0G 1.2M 2.4G 5.1G
Swap: 2.0G 0B 2.0G

 SSH Connection Information:
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
inet 192.168.1.199 netmask 255.255.255.0 broadcast 192.168.1.250
inet6 fe80::1bfc:eea:ef85:1ba0 prefixlen 64 scopeid 0x20<link>
ether 18:eb:0f:93:3e:f4 txqueuelen 1000 (Ethernet)
RX packets 100 bytes 10000 (9.7 KiB)
RX errors 0 dropped 0 overruns 0 frame 0
TX packets 100 bytes 10000 (9.7 KiB)
TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0
lo: flags=73<UP,LOOPBACK,RUNNING> mtu 65536
inet 127.0.0.1 netmask 255.0.0.0
inet6 ::1 prefixlen 128 scopeid 0x10<host>
loop txqueuelen 1000 (Local Loopback)
RX packets 100 bytes 10000 (9.7 KiB)
RX errors 0 dropped 0 overruns 0 frame 0
```

```
TX packets 100 bytes 10000 (9.7 KiB)
TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

 [2024-06-03 12:30:00]
CPU Usage: 22.3%
Memory Usage: 35.7%

 [2024-06-03 12:30:30]
CPU Usage: 24.5%
Memory Usage: 36.2%

 [2024-06-03 12:31:00]
CPU Usage: 19.1%
Memory Usage: 33.8%

 [2024-06-03 12:31:30]
CPU Usage: 85.6%
Memory Usage: 40.1%
Warning: CPU usage exceeds threshold!
Killing process with PID: 1234 due to high CPU usage.

 [2024-06-03 12:32:00]
CPU Usage: 20.9%
Memory Usage: 34.5%

 [2024-06-03 12:32:30]
CPU Usage: 18.4%
Memory Usage: 33.9%

 [2024-06-03 12:33:00]
CPU Usage: 17.8%
Memory Usage: 34.2%
```

```
[2024-06-03 12:33:30]
CPU Usage: 21.5%
Memory Usage: 85.3%
Warning: Memory usage exceeds threshold!
Killing process with PID: 5678 due to high Memory usage.

[2024-06-03 12:34:00]
CPU Usage: 23.2%
Memory Usage: 32.4%
```

دقت می‌کنید که این مثال براساس مقادیر آستانه‌ی مشخص دلخواهی است که تنظیم شده است و شما می‌توانید براساس سیستمی که با آن کار می‌کنید مقادیر معین مطلوبی را برای آنها در نظر بگیرید و مقادیر آنها را ذکر نمائید.

همچنین کدهای Makefile و CMakeLists را به گونه‌ای بنویسید که یک سری اطلاعات مهم همچون آدرس SSH سیستم اولیه و ثانویه و پسورد آن را به همراه آن دو تا درصد (میزان) که اگر میزان مصرف به ترتیب پردازنده و حافظه سیستم ثانویه از آنها فراتر بروند هشدارهای مربوطه را به سیستم اولیه می‌فرستد را در همان زمانی که می‌خواهیم اجرا کنیم از ما بعنوان ورودی بگیرند.

توجه شود که در این تمرین زیبایی و کارکرد صحیح کدها در نمره تاثیر دارند. سطح خوش ذوقی شما در انجام این تمرین حائز اهمیت است. این خوش ذوقی شامل خوش سلیقگی در نشان دادن خروجی و پیغام‌ها در آن و خوش سلیقگی در نوشتن کدها می‌باشد.

توجه مهم: برای این تمرین، در نهایت به جز گزارشی که شامل توضیحات و تصاویر مرحله به مرحله است، باید یک کلیپ کوتاهی از آن تهیه کنید که صحت عملکرد برنامه را نشان دهد. نیازی به توضیحات اضافه در مورد کد و ... در این کلیپ نیست چرا که قطعا شما اینکار را در گزارشکار کرده‌اید. این کلیپ بسیار مهم است و اگر به درستی اجرا و عملکرد برنامه را در آن به طور کامل نشان ندهید گویا کدتان صحیح نیست و اشکال دارد.

توجه شود که مطابق مثالی که برای شما برای نمونه خروجی مورد انتظار آورده شده است، شما باید در گزارش و فیلم خود، حتما حداقل سه حالتی که در حالت اول مقدار استفاده از CPU و Memory از مقادیر آستانه‌شان پایین تر هستند، در حالت دوم برای CPU بالاتر و برای Memory پایین‌تر است و در حالت سوم برای CPU پایین‌تر و برای Memory بالاتر است را نشان بدهید.

۳ تمرین سوم: اسکریپت در Linux

در این تمرین می خواهیم نوشتن اسکریپت در لینوکس را تمرین کنیم. همچنین با مبحث سرویس ها در لینوکس آشنا خواهیم شد. کدهای مربوطه و فایل های خروجی را در کنار گزارش خود ارسال کنید.

(الف) یک اسکریپت shell بنویسید که به سامانه net2 شریف لاگین کند. همچنین ، یک سرویس بنویسید که در ابتدا و پس از هر بار boot شدن این اسکریپت را اجرا کند. در مرحله بعد، اسکریپت را به نحوی تغییر دهید که پس از هر ۱۰ ثانیه، اتصال به سامانه net2 را چک کند و در صورت عدم اتصال، مجددا به این سامانه متصل لاگین کند.

(ب) (امتیازی) web scraping فرآیند استخراج داده‌ها از وب سایت‌ها است. توصیه می شود که برای کاربردهایی نظیر web scraping، از اسکریپت پایتون استفاده شود. با استفاده از اسکریپت پایتون سرویسی بنویسید که در ابتدا و پس از boot شدن، وارد سایت دانشگاه (sharif.ir) شده و عناوین خبری در قسمت "اخبار و اطلاعیه‌ها" را در فایلی با عنوان news.txt ذخیره کند.

راهنمایی: برای استخراج اطلاعات یک صفحه، باید محتوای آن صفحه را دریافت کنید. html و css از جمله زبان های مرسوم برای نوشتن صفحات وب هستند. برای پیدا کردن قسمت موردنظر در صفحه sharif.ir می‌توانید از ابزار inspector کروم استفاده کنید. همچنین توصیه می شود برای پردازش محتوای صفحه، از کتابخانه BeautifulSoup استفاده کنید.

۴ تمرین چهارم: GDB و Expect Script

فایل PrimeSum.c در پوشه تمرین به شما داده شده است. محتوای این فایل، شامل کدی است که یک عدد طبیعی را به عنوان ورودی می‌گیرد و در صورتی که آن عدد به صورت جمع دو عدد اول قابل نوشتن باشد، ترکیب های ممکن برای آن دو عدد اول را پیدا کرده و برای کاربر چاپ می‌کند. با استفاده از gcc این فایل را در مود دیباگ کامپایل کنید.

(الف) یک اسکریپت expect بنویسید که برنامه را سه بار به ترتیب با ورودی های ۲۵، ۵۰ و ۱۰۰ اجرا کند و تمامی زوج مرتب های اول ممکن را که از جمع آن ها عدد مربوطه ساخته می‌شود پیدا کند و به کاربر نشان دهد. همچنین برای هر عدد، تعداد این زوج مرتب ها را گزارش کند.

(ب) همانطور که احتمالا از بررسی نتیجه بخش قبل متوجه شده اید، این برنامه به درستی کار نمی‌کند. یک اسکریپت GDBبنویسید که فایل کامپایل شده را بررسی کند و مشکلات آن را تشخیص دهد. breakpoint های مناسب بگذارید و مقدار متغیرهای مهم و سایر نکات مفید در دیباگ را در فایلی تحت عنوان debug.txt ذخیره کنید. توجه کنید که هدف از این بخش، نوشتن یک اسکریپت GDB برای دیباگ است و نه صرف پیدا کردن خطاهای کد. در نهایت، اسکریپت expect نوشته شده در بخش قبل را بار دیگر اجرا کرده و از درست بودن عملکرد کد مطمئن شوید. کدهای مربوط به اسکریپت و فایل های خروجی را در کنار گزارش خود ارسال کنید.