دانشکده مهندسی برق

تمرین سری سوم

سیستم های نهفته بلادرنگ

طراحان تمرین:

حسین سلیمانی و مهدی رمضانی (بخش عملی)

محمدمحسن چقازردی (بخش تئوری)

استاد:

دکتر ایمان غلامپور

نیم سال دوم 1403-1404

لینوکس نهفته بر روی برد Orange Pi Zero 2 Plus

سوالات بخش تئوری

1. سیستم عامل لینوکس در اغلب موارد به عنوان یک سیستم عامل "نه چندان بلادرنگ" شناخته میگردد. در چه شـرایطی لینوکس را میتوان برای تسـک های بلادرنگ اسـتفاده نمود؟ چه محدودیت هایی با وجود پتچ های *PREEMPT_RT* هنوز برای لینوکس وجود دارد؟ سناریویی را درنظر بگیرید که سیستم اتوماسیون صنعتی باید حداکثر در مدت زمان 5 میلی ثانیه به تغییرات یک ورودی واکنش مناسب نشان دهد. برای این سـناریو اسـتفاده از لینوکس امبدد با پتچ های *PREEMPT_RT* را توصـیه میکنید یا یک سیستم عامل بلارنگ نظیر FreeRTOS؟ دلیل خود را براساس خصوصیات سیستم شرح دهید.
2. رفتار و موارد استفاده سیاسـت های زمان بندی *SCHED_FIFO, SCHED_RR* در لینوکس را با یکدیگر مقایسه کنید. برای یک سیستم داده برداری با خصوصیات بلادرنگ نرم[1]، به چه صورت سیاست زمان بندی مناسب را انتخاب خواهید کرد؟
3. سیستم های امبدد اغلب از حافظه فلش استفاده میکنند. دو فایل سیستم رایج لینوکس در دستگاه های امبدد را با یکدیگر مقایسه کنید. (برای مثال JSFS2, ext4) تفاوت های کلیدی این دو فایل سیستم از نظر میزان سایش[2]، زمان نصب[3] و اطمینان پذیری بعد از قطع تغذیه[4] را با یکدیگر مقایسه نمایید.
4. چه خطرات و آسیب پذیری هایی سیستم های لینوکس امبدد متصل به اینترنت را تهدید میکند؟ یک نمونه را مثال بزنید و برای آن راه حلی ارایه کنید که مصـالحه و تعادلی مناسـبی بین امنیت و عملکرد ارائه دهد.
5. در توسـعه سـیسـتم های امبدد لینوکس بر روی برد هایی نظیر BeagleBone یا Raspberry Pi، مفهومDevice Tree Overlay چیست و به چه صورت در زمان اجرا به صورت پویا پریفرال های سخت افزاری را پیکربندی میکند؟ مثال واقعی شرح دهید.

[1] Soft real-time data acquisition system
[2] Wear leveling
[3] Mount time
[4] Reliability after power loss

بخش عملی

آشنایی اولیه با سیستم عاملLinux

در این تمرین قرار است کمی با بعضی از بخش های سیستم عامل لینوکس آشنایی پیدا کنید. به سوالاتی که در ادامه مطرح می شود تا جایی که ممکن است دقیق و کامل و در عین حال مختصر توضیح دهید. پر واضح است که هر چه (خصوصا بخش های تحقیقاتی) دقیق تر و کامل تر 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 بخش های آن به زبان اسمبلی است. تحقیق کنید که اولا دقیقا آن بخش هایی که به زبان اسمبلی نوشته شده اند 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 باک هایی در سیستم عامل 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 می خواهیم 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 دانشجویی تان و شماره ای که با آن عضو کانال رسمی درس هستید به همراه آدرس جیمیل تان که مرتبط با گوگل درایو تان است (در خطوط جداگانه) سپس آن را مجدد در گوگل درایو خود آپلود نمائید) .آن را Access Openکنید و لینک آن را در گزارش بگذارید (.در گزارش تان حتما تصاویر کافی به همراه توضیحات از تمام مراحل کار باشد .

تمرین CMake – Make :در این تمرین قرار اســت که برنامه یک monitor remote را بنویســید و به درســتی آن را اجرا کنید. کد این برنامه را به زمان C++ می نویســید و برای بهره برداری از آن از Make و CMakeاســتفاده می نمائید .برای انجام این تمرین در بخش برنامه کد C++ باید از پروتکل ارتباطی SSH استفاده کنید به گونه ای که از یک سیستم دارای سیستم عامل لینوکس به یک سیستم دیگر متصل شوید و پس از تبادل یک سری پیغام های 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 می دهد، اگر هر یک از میزان های (درصدهای) استفاده از پردازنده و حافظه در سیستم دوم از آستانه های مشخصی که توسط کاربر در سیستم اول تعیین می شوند فراتر روند، پس از نشان دادن اطلاعات سخت افزاری 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 گونه ای برنامه را بنویســید که به صــورت اتوماتیک هر ۳۰ ثانیه یک بار سیستم اولیه از طریق SSH سخت افزار سیستم ثاونیه را مانیتور کند و پیغام ها و عملیات های لازم که در بالا شرح داده شد را انحام بدهد .سپس باید یک Makefile و CMakeLists استاندارد بنویسید که پروژه را buildکند .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
|
—├Makefile
—├CMakeLists.txt
—├README.md
—└build /
```

البته پر واضح است که این نمایشی که ملاحظه کردید پیش از build شدن پروژه توسط Makefile و CMakeListsاست و شما در فایلی که می فرستید علاوه بر فایل توضیحات، در این پوشه باید پوشه هایی که حاصل از build شدن پروژه ایجاد می شوند نیز موجود باشد. در نتیجه باید در کد های Makefile و CMakeLists به این نکته توجه کنید که اگر آن پوشه ها یا فایل های 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 می تواند به شکل زیر باشد.

```
sudo apt-get update
sudo apt-get install -y openssh-server
sudo systemctl enable --now ssh
SSH connection established and authenticated successfully!
Connecting to remote host...
Connection established.

System CPU Information:
  Architecture: x86_64
  CPU(s): 4
  Model name: Intel(R) Core(TM) i5-8250U CPU @ 1.60GHz
...

System Memory Information:
  total: 7.7G   used: 2.3G   free: 3.0G   buff/cache: 2.4G   available: 5.1G

SSH Connection Information (eth0):
  inet 192.168.1.199  netmask 255.255.255.0  broadcast 192.168.1.255
...

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

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

...

[12:31:30 03-06-2024]
CPU Usage: 85.6%
Memory Usage: 40.1%

Warning: CPU usage exceeds threshold!
Killing process with PID: 1234 due to high CPU usage.
```

دقت می کنید که این مثال براساس مقادیر آستانه ی مشخص دلخواهی است که تنظیم شده است و شما می توانید براساس سیستمی که با آن کار می کنید مقادیر معین مطلوبی را برای آنها در نظر بگیرید و مقادیر آنها را ذکر نمائید .همچنین کد های Makefile و CMakeLists را به گونه ای 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 هشدار های مربوطه را به سیستم اولیه می فرستد را در همان زمانی که می خواهیم اجرا کنیم از ما بعنوان ورودی بگیرند .توجه شود که در این تمرین زیبایی و کارکرد صحیح کد ها در نمره تاثیر دارند. سطح خوش ذوقی شما در انجام این تمرین حائز اهمیت است. این خوش ذوقی شامل خوش سلیقگی در نشان دادن خروحی و پیغام ها در آن و خوش سلیقگی در نوشتن کد ها می باشد .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 کرده اید. 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 Memoryاز مقادیر آستانه شان پایین تر هستند، در حالت دوم برای CPU بالاتر و برای Memory پایین تر است و در حالت سوم برای CPU پایین تر و برای Memory بالاتر است را نشان بدهید.

تمرین دوم : راه‌اندازی برد OrangePi Zero 2 Plus

به تمرین سیستم‌های نهفته لینوکس برای راه‌اندازی و کاوش برد OrangePi Zero 2 Plus خوش آمدید. مراحل زیر را دنبال کنید تا برد را پیکربندی کنید، به شبکه‌ها متصل شوید و وظایف را برای درک عمیق‌تر سیستم‌های نهفته پیاده‌سازی کنید.

ویدئوی آموزشی را دنبال کنید تا برد OrangePi Zero 2 Plus را پیکربندی کنید.

اتصال به Wi-Fi و SSH

با استفاده از گوشی یا لپ‌تاپ خود یک هات‌اسپات Wi-Fi ایجاد کنید با SSID و رمز عبور دلخواه. مطمئن شوید فرکانس روی 2.4 گیگاهرتز تنظیم شده و دسترسی به اینترنت فعال است.

نکته: از دستور nmcli برای اتصال به Wi-Fi استفاده کنید.

نکته: برای جریان کاری روان‌تر، از طریق افزونه Remote – SSH در VS Code یا یک کلاینت SSH (مانند PuTTY) به برد متصل شوید.

اتصال خودکار به Wi-Fi

یک اسکریپت شل با استفاده از nmcli بنویسید تا در صورتی که دستگاه به شبکه متصل نیست، به‌طور خودکار به یک شبکه Wi-Fi مشخص (مانند MySSID) متصل شود.

اسکریپت را با استفاده از cron job طوری تنظیم کنید که هر دقیقه اجرا شود.

اسکریپت را طوری پیکربندی کنید که با استفاده از دستور @reboot در cron، یک‌بار هنگام بوت سیستم اجرا شود.

وضعیت اتصال، شبکه‌های Wi-Fi موجود و سایر جزئیات مرتبط را در فایلی به نام auto-connect.log ثبت کنید.

مکانیزم‌های ارتباط بین‌فرآیندی

سه روش ارتباط بین‌فرآیندی (IPC) را کاوش کنید تا نحوه ارتباط وظایف در لینوکس را درک کنید:

- FIFO
- سیگنال‌های لینوکس
- حافظه اشتراکی

1. FIFO

اسکریپت‌های ارائه‌شده fifo-sender.sh و fifo-receiver.sh در فایل را کامل کنید. هنگام اجرا:

اسکریپت فرستنده باید یک FIFO (در صورت عدم وجود) ایجاد کند و ورودی ترمینال را از طریق FIFO به گیرنده ارسال کند.

گیرنده باید ورودی را فوراً نمایش دهد.

2. سیگنال‌های لینوکس

برنامه‌های C++ ارائه‌شده signal-receiver.cpp و signal-sender.cpp را کامل کنید:

گیرنده: در هنگام راه‌اندازی PID خود را چاپ می‌کند، منتظر سیگنال‌ها می‌ماند و هر سیگنال دریافت‌شده را در یک فایل ثبت می‌کند.

فرستنده: PID و شماره سیگنال را به‌عنوان آرگومان‌های خط فرمان می‌پذیرد و سیگنال را به فرآیند مشخص‌شده ارسال می‌کند.

به سؤالات زیر پاسخ دهید:

1. تفاوت بین سیگنال‌های SIGINT و SIGKILL چیست؟
2. کدام سیگنال‌ها می‌توانند توسط یک فرآیند دریافت یا مدیریت شوند؟
3. چند سیگنال در لینوکس تعریف شده‌اند؟

3. حافظه اشتراکی

برنامه‌های c++ ارائه‌شده shm-writer.cpp و shm-reader.cpp را کامل کنید:

نویسنده: یک قطعه حافظه اشتراکی ایجاد می‌کند و ورودی کاربر را در یک بافر دایره‌ای 10 کاراکتری می‌نویسد.

خواننده: محتوای حافظه اشتراکی را هنگام اجرا می‌خواند و نمایش می‌دهد.

به سؤال زیر پاسخ دهید:

تفاوت‌های کلیدی بین FIFO و حافظه اشتراکی برای IPC چیست؟

وب‌سرور پایه

اسکریپت شل ارائه‌شده را کامل کنید تا یک وب‌سرور ساده ایجاد شود که روی پورت 80 گوش کند و به درخواست‌های HTTP با تاریخ و زمان فعلی سیستم پاسخ دهد.

هنگام اجرا در ترمینال، پاسخ را در ترمینال نیز چاپ کند.

یک سرویس Systemd (webserver.service) ایجاد کنید که:

پس از آنلاین شدن شبکه (اتصال به Wi-Fi) شروع شود.

در صورت خروج اسکریپت، به‌طور خودکار با تأخیر 5 ثانیه‌ای راه‌اندازی مجدد شود.

خروجی و خطاها را در ژورنال سیستم ثبت کند.

هنگام بوت سیستم اجرا شود. فایل سرویس باید در کنار اسکریپت وب‌سرور در ارسال و مخزن قرار گیرد.

نکته: از دستور nc (netcat) برای مدیریت اتصالات شبکه استفاده کنید.

دستورالعمل ارسال تکلیف در CW (اجباری)

در قسمت اول عملی و دوم عملی برای هر بخش اول تمرین یک فیلم از کارکرد کد مطابق توضیحات تهیه کرده :

قسمت اول : 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 کرده اید. 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 آستانه شان پایین تر هستند، در حالت دوم برای CPU بالاتر و برای Memory پایین تر است و در حالت سوم برای CPU پایین تر و برای Memoryبالاتر است را نشان بدهید.

قسمت دوم :

به جز گزارشی که شامل توضیحات و تصاویر مرحله به مرحله است ، یک ویدئو ضبط کنید که مراحل کد شما (مانند توسعه یا اجرا) را نشان دهد

فایل ZIP را در CW آپلود کنید. به سؤالات تکلیف در گزارش کار خود پاسخ دهید.

دستورالعمل ارسال تکلیف فقط برای تمرین دوم قرار دادن ریپازیتوری در گیتهاب (امتیازی)

مخزن خود را به حساب GitHub خود ارسال کنید و آن را عمومی کنید. من مخزن را کلون کرده و بر اساس محتوای آن نمره شما را تعیین می‌کنم. همچنین، یک ویدئو ضبط کنید که مراحل کد شما (مانند توسعه یا اجرا) را نشان دهد. یک فایل متنی به نام repolink.txt ایجاد کنید که فقط شامل آدرس URL مخزن GitHub شما باشد. ویدئو، پوشه مخزن و فایل repolink.txt را در یک فایل ZIP به نام RTES2025_HW3_{شماره_دانشجویی}.zip بسته‌بندی کنید، که {شماره_دانشجویی} شماره دانشجویی شماست. فایل ZIP را در CW آپلود کنید. به سؤالات تکلیف در فایل README.md مخزن با استفاده از سینتکس Markdown پاسخ دهید.

ساختار فایل ZIP باید به این صورت باشد:

```
RTES2025_HW3_401234567
├── HW3.mp4  # فرمت‌های پشتیبانی‌شده: mp4، mkv، mpeg یا مشابه
├── repolink.txt       # شامل آدرس URL مخزن GitHub
└── RTES2025_HW3       # پوشه مخزن
    ├── auto-wifi-connect
    ├── ipc
    └── ...
```

وب‌سایت‌های مرجع مفید

- **OrangePi Documentation**: Official OrangePi Wiki – Guides for setting up OrangePi boards.
- **Linux Networking**: nmcli Documentation – Manage Wi-Fi with `nmcli`.
- **Git Basics**: Git SCM – Official Git documentation for cloning and managing repositories.
- **Systemd Services**: Systemd Documentation – Guide to creating systemd services.
- **Linux IPC**: Beej's Guide to IPC – Comprehensive resource on FIFO, signals, and shared memory.
- **Netcat Usage**: Netcat Manual – Learn how to use `nc` for networking tasks.