به نام خداوند جان و خرد

احسان قاسمی — ۹۸۱۰۲۱۰۸

تمرین اول درس سیستم های نهفته بی درنگ

استاد درس: دکتر غلامپور

یکشنبه؛ ۱۴۰۲/۱/۱۳

بخش اول – تئوری و تحقیق

۱- بخش های مختلف تشکیل دهنده یک سیستم نهفته.

Power supply

تقریبا میتوان منبع تغذیه را به عنوان حیاتی ترین قسمت از یک سیستم نهفته معرفی کرد. معمولا سیستم های نهفته نیازمند ۵ ولت ورودی هستند که میتواند در بازه ۱.۸ ولت تا ۳.۳ ولت باشد. در سیستم های نهفته منبع تغذیه میتواند یک باتری یا حتی یک آداپتور دیواری باشد. منبع تغذیه بر اساس نیاز کاربر و نیازهای برنامه انتخاب میشود. منبع تغذیه باید صاف و کارآمد باشد تا بتوان منبع تغذیه مداوم را به یک سیستم تعبیه شده ارائه کرد. منبع تغذیه نیز باید اجازه اتلاف را بدهد و تا حد امکان کارآمد باشد.

Processor

برای هر سیستم نهفته، پردازنده به عنوان مغز سیستم عمل می کند. پردازنده مسئول تصمیم گیری در مورد عملکرد سیستم نهفته است. در بازار انواع مختلفی از پردازنده ها موجود است و می توان آنها را بر اساس نیاز کاربر انتخاب کرد. سیستم نهفته می تواند به عنوان یک میکروکنترلر و ریزپردازنده عمل کند. این پردازنده می تواند یک پردازنده ۸ بیتی، یک پردازنده ۱۶ بیتی و یک پردازنده ۳۲ بیتی باشد. هر چه بیت کمتر باشد، برنامه برای سیستم های نهفته کوچکتر است. هنگامی که برنامه های کاربردی بزرگ استفاده می شود، پردازنده بیت بالاتر در سیستم تعبیه شده مورد نیاز است. پردازنده باید بسیار سریع باشد، قیمت باید حداقل باشد، عملکرد باید خوب باشد تا بتوان عملکردها را در یک سیستم تعبیه شده بسیار سریع انجام داد.

Memory

از آنجایی که میکروکنترلرهای مختلفی در سیستم نهفته استفاده می شود، حافظه در خود میکروکنترلر وجود دارد. اساسا دو نوع حافظه RAM (حافظه دسترسی تصادفی) و ROM (حافظه فقط خواندنی) وجود دارد. از آنجایی که RAM از نوع حافظه فرار است، داده ها می توانند به طور موقت در حافظه ذخیره شوند و هنگامی که سیستم خاموش می شود، داده ها از حافظه از بین می روند. حافظه فقط خواندنی به عنوان حافظه کد طبقه بندی می شود. ROM برای ذخیره برنامه استفاده می شود و زمانی که سیستم روشن است، کدهای سیستم تعبیه شده را از حافظه ROM واکشی می کند.

Timers counter

در برخی از برنامه ها همیشه نیاز به تاخیر وجود دارد که باید در برنامه ارائه شود. به عنوان مثال، در برنامه های نمایش LED نیاز به تأخیر وجود دارد تا LED بتواند به چشمک زدن ادامه دهد. و برای آن می توان از تایمر و شمارنده در سیستم نهفته استفاده کرد. برنامه نویسی را می توان به گونه ای انجام داد که تأخیر بتواند سیستم نهفته را تولید کند. بازه زمانی تاخیر را می توان با استفاده از نوسانگر کریستالی و فرکانس سیستم تعیین کرد تا بتوان بر اساس نیاز کاربر تاخیر ایجاد کرد.

Communication ports

پورت ارتباطی نوعی رابط است که برای ارتباط با انواع دیگر سیستم های نهفته استفاده می شود. در سیستم نهفته انواع مختلفی از درگاه های ارتباطی مانند RS-485، Ethernet، USB، UART و بسیاری دیگر وجود دارد. هنگامی که یک سیستم تعبیه شده در مقیاس کوچک استفاده می شود، درگاه های ارتباطی می توانند از میکروکنترلر استفاده شوند. همچنین پروتکل های سریالی وجود دارد که می تواند برای ارسال داده ها از یک برد سیستم به برد دیگر استفاده شود.

Outputs and inputs

هنگامی که از سیستم نهفته استفاده می شود، ورودی برای تعامل با سیستم مورد نیاز است. ورودی سیستم نهفته می تواند توسط سنسور یا توسط خود کاربر ارائه شود. پردازنده مورد استفاده در سیستم نهفته می تواند بر اساس ورودی و خروجی باشد. برای استفاده از پورت ورودی و خروجی باید پیکربندی مناسب انجام شود. در سیستم نهفته پورت های ورودی و خروجی ثابتی وجود دارد تا دستگاه ها فقط به آن پورت های مشخص شده متصل شوند. به عنوان مثال، P0، P1، P2، و بسیاری دیگر.

Circuits used in application

هنگامی که سیستم نهفته طراحی می شود، اجزای سخت افزاری مختلفی وجود دارد که می توانند برای اهداف طراحی مورد استفاده قرار گیرند. انتخاب مدار کاملاً به کاربرد مورد استفاده برای سیستم های نهفته بستگی دارد. به عنوان مثال، در کاربردهای سنسور دما، نیاز به سنسورهای دما برای اندازه گیری دما وجود دارد.

۲- پنج زبان برنامه نویسی رایج در خصوص طراحی یک سیستم نهفته و سخت افزار هدف آنها.

C Programming	Ada
Embedded C	Assembly
C++	Rust
Python	Verilog

Language	Characteristics	Pros	Cons	Typical Uses
C	Compiled language. Simple and very widely used. A building block of many other languages.	Highly efficient programming language. Stable. Fast.	Low-level language, which means it can be difficult to use due to technical details a programmer must remember. It can lead to developers creating problems as they work on it. Modern coding techniques are challenging to implement within the language.	Used in many embedded systems.
C++	A compiled language. It has most or all elements of C and other capabilities that C doesn't have.	It has a powerful standards library, which saves programmers time in writing code. C++ can be as efficient as C.	Complex language that can be difficult to learn.	Used in many embedded systems.
Python	Interpreted language.	Developers don't have to cross-compile code for embedded systems. Quite easy to learn. It makes writing to and reading from hardware easy.	Not deterministic, not for real-time operating systems. Lacks a compiler and static analyzers that might find problems during compilation. It must be tested well to avoid runtime errors.	Used in application processors and as a communication vehicle between the user and the embedded system.
MicroPython	An implementation of Python optimized to run on microcontrollers.	Open source. It has libraries built in to support many tasks.	Code isn't as fast and might use more memory compared to similar low-level C/C++-based code. Code with tight timing or performance requirements might not work in MicroPython. It will be a bit slower because it has to interpret every instruction and convert it to CPU code.	Used in embedded systems with microcontrollers.
Java	An efficient, general-purpose language used extensively for internet-based applications.	Once written in an embedded system, code is very portable to another device. Reliable.	Uses "garbage collectors" which can lead to performance issues, including with GUIs. It can't be used in real-time systems.	Used in embedded systems running on an Android operating system.
JavaScript	A text-based programming language used in some embedded systems.	A good solution if your system requires significant networking and graphics and uses HTML5.	Poor runtime efficiency. Not easy to maintain.	Especially used in systems with a human-machine interface.
Ada	Created in the 1970s as a U.S. Department of Defense Project because the department was concerned with the hundreds of programming languages in its embedded systems.	Extremely efficient and reliable.	Can be challenging to learn. Not widely used.	Used in systems where safety is critical, and you must formally verify the system works correctly. Used in military systems, missiles, trains, and medical devices.
Assembly	Provides basic low-level access that manipulates hardware efficiently.	Memory efficient and fast.	It's complicated to read and write.	Used mostly for high-performance algorithms that require hand-based fine-tuning.
Rust	Created by an employee of Mozilla and released in 2010.	It has many modern programming language capabilities and security features. It helps encourage secure code, with fewer bugs.	Not standardized and not used very much yet. Takes time to compile.	It will take time before it finds more uses in embedded systems.

۳- سه روش متفاوت حمله به یک سیستم نهفته. به طور خاص حمله Buffer Overflow Attack.

"The only truly secure system is one that is powered off, cast in a block of concrete, and sealed in a lead-lined room with armed guards." — Gene Spafford

ما نمی توانیم همانطور که محقق بزرگ علوم کامپیوتر در بالا گفته است انجام دهیم. اما ما می توانیم انواع رایج ترین حملات به سیستم های نهفته را شناسایی کرده و خطرات آنها را کاهش دهیم.

Software-based Attacks

هکرها می توانند به راحتی به نرم افزار حمله کنند زیرا به دانش خاصی نیاز ندارند. در عوض، آن‌ها می‌توانند از راه دور با استقرار بدافزار، اعمال خشونت‌آمیز، یا سوءاستفاده از آسیب‌پذیری‌های امنیتی برنامه‌های وب به نرم‌افزار حمله کنند. علاوه بر این، تمرکز بیشتری بر روی ایمن سازی اجزای فیزیکی سیستم وجود دارد. هکرها می توانند به راحتی به داده های حساس دسترسی پیدا کنند یا از طریق این روش کنترل سیستم نهفته را به دست آورند.

Network-based Attacks

این حمله را می توان از راه دور نیز انجام داد. هکرها از آسیب پذیری های زیرساخت شبکه سوء استفاده می کنند و می توانند ترافیک منتقل شده توسط سیستم تعبیه شده را بدست آورند و اصلاح کنند. رایج ترین حملات شامل پارازیت سیگنال، ربودن جلسه، مسمومیت سیستم نام دامنه (DNS) و حملات انسان در وسط (MITM) است.

Side-channel Attacks

حمله کانال جانبی گران‌ترین حمله است، زیرا به هکر نیاز دارد که دانش تخصصی در مورد اجزای سخت‌افزاری داشته باشد. اما اگر هکر در انجام این کار موفق باشد، می‌تواند اطلاعاتی در مورد مصرف برق، نشت الکترومغناطیسی، زمان‌بندی عملیات و غیره به دست آورد.

و در قسمت بعد هم درباره buffer overflow attack بررسی خواهیم کرد.

buffer overflow attack نوعی حمله است که در آن مهاجم اطلاعات بیشتری را به بافر می فرستد تا بتواند از عهده آن برآید و باعث سرریز شدن داده های اضافی به مکان های حافظه مجاور می شود. این کار می تواند باعث رفتار غیرمنتظره در برنامه شود، از جمله خرابی و اجرای کد دلخواه.

در زمینه سیستم های تعبیه نهفته، حملات سرریز بافر می تواند به ویژه خطرناک باشد، زیرا بسیاری از سیستم های نهفته منابع محدودی دارند و ممکن است سطح ویژگی های امنیتی مشابه سیستم های بزرگتر را نداشته باشند. یک مهاجم می‌تواند به طور بالقوه از آسیب‌پذیری سرریز بافر در یک سیستم نهفته برای اجرای کدهای مخرب، سرقت اطلاعات حساس یا ایجاد اختلال در سیستم سوء استفاده کند.

برای جلوگیری از حملات سرریز بافر، طراحی دقیق نرم افزار و استفاده از تکنیک های برنامه نویسی مناسب، مانند بررسی کرانه ها و اعتبارسنجی ورودی، مهم است. علاوه بر این، به روز نگه داشتن نرم افزار با وصله های امنیتی و محدود کردن دسترسی به سیستم ها و داده های حساس حایز اهمیت خواهد بود.

۴- پنج پروتکل ارتباطی رایج برای اتصال به یک سیستم نهفته. به طور خاص UART.

UART Protocols

UART مخفف Universal Asynchronous Receiver Transmitter (UART) است و شاید ساده‌ترین، قدیمی‌ترین و رایج‌ترین شکل ارتباط دستگاه به دستگاه باشد. همانطور که از نام آن پیداست، UART از ارتباط سریال ناهمزمان استفاده می کند، به این معنی که هیچ سیگنال ساعتی برای همگام سازی داده ها بین فرستنده و دستگاه های گیرنده استفاده نمی شود. بنابراین، برخلاف اکثر پروتکل ها، انتقال داده UART از TX-RX پیروی می‌کند.

SPI Protocols

SPI به رابط جانبی سریال اشاره دارد و یک پروتکل ارتباطی معروف در دنیای جاسازی شده است. پروتکل SPI در طیف وسیعی از چیپست ها، از جمله ماژول های کارت SD و کارت خوان RFID استفاده می شود. وظیفه اصلی آن به عنوان یک گذرگاه واسط ارسال است.

I2C Protocols

I2C به معنای مدار یکپارچه کارآموز است و اولین بار توسط فیلیپس برای تامین برق روشنایی الکتریکی ایجاد شد. ارتباطات I2C به دلیل ساختار Multi-Master-Multi-Slave آن، که به نام I2C Bus نیز شناخته می شود، محبوب است. چنین ساختاری هنگام ثبت داده های چند میکروکنترلر برای تامین انرژی یک سیستم دستگاه ارزشمند است. I2C کاربردهای متفاوتی دارد، به ویژه در مدارهای آی سی مانند ماژول های LCD، سنسورهای دما، و اکتساب داده و غیره، که آن را به یک پروتکل رایج مورد استفاده توسط مهندسان تعبیه شده تبدیل می کند. پروتکل ارتباطی I2C از پیکربندی Half-Duplex پیروی می کند، به این معنی که داده ها را می توان ذره ذره از طریق

ارتباط دو طرفه در یک زمان واحد منتقل کرد. این باید هنگام فکر کردن به یکپارچه سازی در نظر گرفته شود، زیرا ارتباطات داده با سرعت کمتری انجام می شود. مانند SPI و I2C ،UART از یک پروتکل ارتباطی سریالی پیروی می کند که امکان ارتباط رابط دو سیمه را بین Masters و Slave فراهم می کند. با داشتن ساختار پین کمتر پیچیده، I2C برای ادغام جذاب است. در یک گذرگاه I2C، ارتباط چند دستگاه Master و Salve از طریق ارتباط همزمان کار می‌کند، جایی که یک سیگنال ساعت کنترل‌شده توسط Master بین گره‌های salve در میان رابط‌های دو سیم، خط ساعت سریال (SCL) و خط داده سریال (SDA) توزیع می‌شود. . هنگام ادغام موارد استفاده از I2C باید مورد ارزیابی قرار گیرد زیرا پیچیده تر از بسیاری از پروتکل های دیگر است.

USB Protocols

USB به گذرگاه سریال جهانی اشاره دارد و از یک پروتکل بین سیستمی پیروی می کند که بین دو دستگاه ارتباط برقرار می کند. برجسته در جهان دستگاه های الکترونیکی کامپیوتر، USB بیش از پروتکل های UART محبوب شده است. USB از یک پروتکل سریال ناهمزمان پیروی می کند که در آن به سیگنال ساعت نیازی نیست و آن را به دستگاهی کم هزینه تبدیل می کند. هنگامی که دستگاه میزبان ارتباط ارسال می شود، انتقال داده از طریق بسته های داده به دستگاه نقطه پایانی دریافت کننده رله می شود. USB می‌تواند طیف وسیعی از سرعت‌ها را بسته به مورد استفاده، از ۱.۵ مگابایت تا ۱۰ گیگابایت داشته باشد.

CAN(Controller Area Network)

این پروتکل معمولا در کاربردهای خودروسازی و صنعتی استفاده می شود. این امکان برقراری ارتباط قابل اعتماد بین دستگاه ها را در فواصل طولانی فراهم می کند و می تواند چندین دستگاه متصل به یک اتوبوس واحد را اداره کند.

۵- دست کم یک مورد از اشتباهات ممکن در طراحی یک سیستم نهفته که منجر به خطرات ایمنی میشود.

Inadequate safety features

برخی از سیستم‌های تعبیه‌شده فاقد ویژگی‌های ایمنی کافی مانند سنسورها یا مکانیسم‌های ایمن هستند که می‌تواند منجر به نقص یا تصادف شود.

Inadequate testing

سیستم های تعبیه شده برای اطمینان از ایمنی و قابلیت اطمینان آنها نیاز به آزمایش دقیق دارند. با این حال، برخی از طراحان ممکن است آزمایش کافی را انجام ندهند، که منجر به اشکالات یا خطاهای شناسایی نشده ای می شود که می تواند منجر به خرابی سیستم شود.

Insufficient redundancy

عدم وجود افزونگی در سیستم های بحرانی می تواند باعث خرابی های خطرناک شود. به عنوان مثال، اگر منبع تغذیه پشتیبان خراب شود، می تواند منجر به از دست دادن کنترل یک سیستم حیاتی ایمنی شود.

Inadequate security

آسیب پذیری های امنیتی در سیستم های تعبیه شده می تواند خطرات قابل توجهی ایجاد کند. به عنوان مثال، اگر یک سیستم کنترل کننده یک نیروگاه هک شود، می تواند منجر به یک شکست فاجعه بار شود.

Overreliance on software

اتکای بیش از حد به نرم افزار می تواند منجر به شکست در سیستم های تعبیه شده شود. به عنوان مثال، اگر نرم افزار کنترل کننده یک سیستم حیاتی ایمنی به درستی طراحی، آزمایش یا نگهداری نشود، می تواند منجر به خرابی سیستم شود.

Insufficient documentation

فقدان مستندات کافی می تواند منجر به مشکل در نگهداری، تعمیر یا ارتقاء سیستم های تعبیه شده شود که می تواند منجر به خرابی سیستم شود. ضروری است که اطمینان حاصل شود که سیستم های تعبیه شده با ایمنی، قابلیت اطمینان و امنیت طراحی شده اند تا از هرگونه خرابی خطرناک جلوگیری شود.

۶- علت دقیق بروز سانحه انفجار موشک Ariane 5.

انفجار موشک آریان ۵ در سال ۱۹۹۶ به دلیل یک خطای نرم افزاری در سیستم هدایت موشک ایجاد شد. سیستم مرجع اینرسی موشک (IRS) - که جهت گیری و شتاب موشک را اندازه گیری می کند - داده هایی تولید می کند که بیش از بردی است که نرم افزار هدایت موشک برای کنترل آن برنامه ریزی شده بود. این خطا باعث شد که کامپیوتر آنبورد موشک فرمان خاموش شدن اضطراری را صادر کند که به نوبه خود باعث خاموش شدن موتورهای اصلی موشک شد. سپس موشک به دلیل نیروهای آیرودینامیکی شروع به شکستن کرد و در نتیجه وسیله نقلیه و محموله های آن به طور کامل از بین رفت. IRS در موشک آریان ۵ بر اساس سیستم مورد استفاده در موشک آریان ۴ بود. با این حال، موشک آریان ۵ موتور اصلی قوی تری داشت که باعث شد موشک شتاب بالاتری نسبت به آریان ۴ تجربه کند. این به نوبه خود باعث شد که IRS داده هایی تولید کند که از بردی که نرم افزار هدایت برای کنترل آن برنامه ریزی شده بود فراتر می رفت. . این خطا به یک ماژول نرم افزاری خاص به نام ماژول سیستم مرجع اینرسی-نگرش و سیستم مرجع سرعت (IRS-AVS) ردیابی شد. این

نرم افزار در ابتدا برای موشک آریان ۴ توسعه داده شد و در آریان ۵ بدون تغییر برای شتاب بیشتر موشک جدید مورد استفاده مجدد قرار گرفت. این حادثه اهمیت توسعه و آزمایش نرم افزار دقیق در سیستم های حیاتی ایمنی را برجسته می کند. همچنین نیاز به ارزیابی و آزمایش کامل اجزا و سیستم‌های فرعی را هنگام تطبیق آنها با سیستم‌های جدید نشان می‌دهد، حتی اگر در برنامه‌های قبلی ثابت شده باشند.

۲- دادهای ارائه شده درجدول زیر را در نظر بگیرید دو پردازندهA,B به ترتیب به فرکانس های کاریGHz 2GHz, 1.2 در دسترس است. میخواهیم برنامه ای با 10M دستور را روی A و برنامه ای با 6M دستور را روی B اجرا کنیم.

- **زمان اجرای هر برنامه را برای هر دو حالت محاسبه کنید.**
- **عملکرد هر پردازنده برای هر حالت چند MIPS است؟**

برای این کار ابتدا کافی است که بررسی کنیم در هر دو حالت در صورتی که 1M دستور داشته باشیم چند clock طول خواهد کشید.

$$first\ state: 1M\times(0.45\times3+0.25\times5+0.18\times2+0.12\times2)=3.2\,M$$

$$second\ state: 1M\times(0.45\times3+0.25\times3+0.18\times1+0.12\times6)=3M$$

حال کافی است این دو مقدار را در عکس فرکانس ضرب کرده تا از کلاک به زمان تبدیل کنیم.

	First program	Second program
First satae	$10\times3.2\times1/2G=0.016s$	$6\times3.2\times1/1.2G=0.016s$
Second state	$10\times3\times1/2G=0.015s$	$6\times3\times1/1.2G=0.015s$

برای مقایسه کردن عمل‌کرد هر پردازنده در هر حالت کافی است همان جدول بالا را این بار بدون لحاظ کردن برنامه‌ها در نظر بگیریم.

	First program	Second program
First satae	$3.2\times1/2G=0.0016$	$3.2\times1/1.2G=0.0026$
Second state	$3\times1/2G=0.0015$	$3\times1/1.2G=0.0025$

بخش دوم – شبیه سازی رایانه ای

تمرین اول)

فایل کامل این تمرین را که کدهای ناقص فرستاده شده را در آن تکمیل کرده ام هم در قالب یک فایل .py , و هم یک فایل ipynb است را در پیوست فرستاده ام که به ترتیب دارای کدها و نیز نتایج شبیه سازی برای هر بخش میباشد. با توجه به کامنت گذاری ای که کرده ام تمامی کدها را میتوانید بدون هیچ ابهامی بررسی کنید.

تمرین دوم)

این تمرین را هم که امتیازی بود در کنار تمرین قبلی زده ام و در فایل های اصلی قابل مشاهده است.