تاریخ انتشار: ۲۶ آذر ۱۴۰۲

اصطلاحاتی از مقاله Reliable Real-Time Operating System for IoT Devices

نویسنده: علیرضا سلطانی نشان، دانشجوی ارشد مهندسی نرم‌افزار – دانشگاه آزاد اسلامی واحد تهران_ شمال a.soltani@iau-tnb.ac.ir

فهرست مطالب

۱ مقدمه

یک پاراگراف مناسب برای مقدمه این برگه نیاز به نگارش است.

موضوعی که در ابتدا مطرح میکنه، در مورد فراگیر شدن گسترده دستگاه‌های مبتنی بر IoT هستش که میگه از وسایل خانه گرفته تا مهم‌ترین وسایل پزشکی. به طوری که به صورت گسترده در زندگی انسان‌ها در حال پیشرفت می‌باشند.

نتیجه این برگه به طور کلی، ارزیابی محققان را بر سیستم عامل TizenRT نشان می‌دهد که تسک‌هایی که حاوی خطا هستند را از فضای رم جدا نگهداری می‌کند در حالی که تضمین اجرای بدون مشکل را برای تسک‌های Real-Time به صورت کامل می‌دهد که در مدت زمان معینی که قرار است یک تسک کامل شود، انجام گیرد (در اینجا بهترین زمان برای انجام تسک را ۵۰ میکروثانیه دیده اند). در ادامه به آن می‌پردازد، تسکی به خاطر خطا متوقف شد چگونه می‌تواند به چرخه حیات مجدد خودش باز گردد؟ معرفی ویژگی Recovery Fast از این سیستم عامل نشان دهنده آن است که بدون نیاز به Reboot کردن سیستم عامل می‌تواند تسک مشک دار قبلی را در مرحله اجرای مجدد قرار داد (بهترین زمانی که محققان برای ارزیابی در نظر گرتفن ۱۰ میلی ثانیه بوده است). به این دلیل است که سیستم عامل TizenRT را انتخابی برای ماموریت‌های خاص (انجام تسک‌های حساس، مهم و بحرانی) معرفی می‌کند.

این مقاله به طور کلی به دو مورد از ویژگی‌های اصلی که یک سیستم عامل Time Real می‌پردازد:

ضعف اصلی برنامه نویس به دلیل پیچیدگی (در محیط و اشل‌های گسترده) نرم‌افزار می‌باشد.

۱.۱ ویژگی‌ها

۱.۱.۱ ویژگی Fault isolation

ویژگی Fault isolation از اسمش معملومه، یعنی جداکننده خطا و فاجعه نرم‌افزاری یک برنامه از دیگر برنامه‌ها. اگر یک برنامه دچار خطا شود، سیستم عامل آن را به صورت خودکار از برنامه‌های دیگر جدا می‌کند تا این حادثه بر اثر خرابی یک برنامه، روی برنامه‌های دیگر تاثیر نگذارد. دلیل اصلی این ویژگی حضور Per-binary Memory Protection هستش که باید تو این بین بررسی بشه. در حقیقت مهمترین قابلیت این ویژگی جلوگیری از عمل راه‌اندازی مجدد یا Rebooting است. (احتمالا توی مقاله منظور از binary User اون نرم‌افزارهایی هستش که برنامه نویس در مد کاربر اونا رو اجرا میکنه). توابع Fault handler بالاترین اولویت را در راه‌اندازی هاThread دارند. برای داشتن همچین قابلیتی نیازمند آن هستیم که قابلیت‌های Real-Time را در سیستم به ذاتی داشته باشیم (یا حتی به وجود آورده باشیم).

۲.۱.۱ ویژگی Fast Recovery

در مقابل ویژگی به نام Fast recovery وجود دارد که به برنامه کمک می‌کند در مدت زمانی بسیار معقول و سریع، برنامه‌ای که با شکست مواجه شده است را ریلود و مجددا اجرا کند که بتواند به ادامه فرایند محاسباتی خودش بپردازد. مکانیزمی که برای Fast recovery پیاده‌سازی شده است که از مرتبه و اولویت پایین‌تری نسبت به هایReal-Time Tread برخوردار است. این عملیات به گونه‌ای انجام می‌شود که عملکرد برنامه‌های حساس دیگر را تحت تاثیر قرار ندهند.

۲.۱ موضوع Per-binary Memory Protection

در این قسمت صد درصد مطمئن شدم که منظور از Binary همون Executable Programها می‌باشد. قابلیتی در سیستم عامل‌ها و پردازنده‌های مدرن و امروزی است که به برنامه‌ها اجازه میدهند که به صورت انفرادی دسترسی به مموری خودشان داشته باشند و آن را به صورت کاملا مستقل کنترل و محافظت کنند. این بدان معناست که هر برنامه در حال اجرا می‌تواند مجموعه‌ای از دسترسی‌ها و محدودیت‌های منحصر به فرد خودش را داشته باشد. مثلا تا چه حدی می‌تواند به حافظه خودش دسترسی داشته باشد. اگر برنامه‌ای تلاش کند که به مجوزی که برای اون نیست دسترسی داشته باشد از آن جلوگیری می‌شود. این قابلیت باعث می‌شود تا برنامه روی مموری‌های یکدیگر دخالت نداشته باشند. این نوع محافظت از حافظه، از مهمترین قابلیت‌های امنیتی در کامپیوتر است، زیرا از نفوذ بدافزارها و آسیب پذیری‌هایی که از طریق دسترسی به حافظه عمل می‌کنند، جلوگیری می‌کند.

۳.۱ اجرای همزمان تسک‌های RT و NRT

سیستم عامل TizenRT می‌تواند تمام تسک‌های RT و NRT را با توجه به دو ویژگی ایزوله‌سازی خطا و بازیابی سریع، به صورت همزمان اجرا کند. در جدول ۱ می‌توانید به تفکیک ۳ معیار تسک‌های RT را با NRT مقایسه کنید.

جدول ۱: Reliable Real-Time Operating System for IoT Devices تسک‌ها

نوع تسک	کد	کاربرد	پایداری
تسک‌های بلادرنگ (RT Tasks)	ساده و کم	موتورهای الکتریکی، کنترل فن‌ها	ساده و پایدار
تسک‌های Non-real time (NRT) Tasks	پیچیده و بزرگ	IoT (OCF, MQTT, TLS, Wi-Fi, BLE)	کاملا مستعد به خطا هستند

محققان آزمایشاتی به منظور بررسی عملکرد یک سیستم عامل IoT در شرایط دشوار انجام دادند. در این آزمایشات، یک Thread از نوع Real-Time در یک برنامه، هر ۵۰ میکروثانیه یک وقفه خارجی را پردازش می‌کند و یک Thread از نوع NRT در یک برنامه دیگر عمدا یک خطای حافظه ایجاد می‌کند. این آزمایشات نشان می‌دهد که Thread نوع RT با موفقیت تسک‌های خود را هر ۵۰ میکروثانیه انجام می‌دهد حتی در حالی که برنامه موجب خطای حافظه شده باشد، و می‌توان نتیجه گرفت که سیستم عامل توانایی بازیابی از خطا را داراست. مهمترین نکته در این میان وقوع وقفه‌ها هر ۵۰ میکروثانیه است که می‌تواند برای ارزیابی ویژگی قابل اعتماد بودن IoTOS در شرایط دشواری محسوب شود [۱].

Fast Interrupt Notification ۴.۱

یک روشی است که ممکنه به منظور اعلام سریع از وقوع یک نقص یا خطا در سیستم‌های کامپیوتری باشد. این امر می‌تواند باعث جبران تاخیرهای مربوط به جداسازی خطا و بازیابی سریع گردد.

۵.۱ فشرده‌سازی باینری‌ها

روشی برای کاهش حجم داده‌های باینری است، با کمک این روش، زمان انتقال داده‌ها از طریق اتصالات بی‌سیم مانند WiFi و Bluetooth و همچنین حافظه‌های ذخیره‌سازی، کاهش می‌یاد. درست است که با کم حجم کردن باینری‌ها باعث انتقال سریع آنها می‌شود اما مهم‌ترین اتفاقی که رخ می‌دهد سپری شدن زمان بیشتر برای فرایند فشرده‌سازی است. برای مثال وقتی می‌خواهیم یک باینری ۲ مگابایتی را با نسبت ۳/۳۴ فشرده‌سازی کنیم و سپس اقدام به ارسال آن کنیم، زمان لودینگ ۳۲ درصد افزایش پیدا می‌کند. در حالت کلی اگر زمان فشرده‌سازی صرفه هزینه‌ای داشته باشد، استفاده از مکانیزم فشرده‌سازی کاملا مناسب خواهد بود.

۶.۱ واحد محافظت از حافظه

یک واحد سخت افزاری [۱] در برخی میکروکنترلرها و پردازنده‌هاست که به برنامه نویسان این امکان را می‌دهد که دسترسی به حافظه را مدیریت و کنترل کنند. به واسطه این واحد می‌توان بخش‌هایی از حافظه را به صورت مجزا نگهداری کرد به گونه‌ای که محدودیت دسترسی به هر بخش از آن حافظه مجزا را تنظیم کرد. این واحد می‌تواند از نفوذ و حمله‌های امنیتی مبتنی بر دسترسی به حافظه جلوگیری کند.

سیستم عامل‌هایی که از این واحد کنترلی پشتیبانی می‌کنند معمولا به صورت Source Open هستند به گونه‌ای که می‌توان به موارد زیر اشاره کرد:

- OS Mbed
- FreeRTOS
- Zephyr

سیستم عامل Mbed دو نوع محافظت پایه‌ای از حافظه را ارائه می‌دهد:

۱. جلوگیری اجرا از RAM: Preventing execution from RAM

۲. جلوگیری از نوشتن روی Flash Memory

این دو ویژگی به صورت خودکار روی سیستم عامل فعال هستند یا اینکه براساس موقعیتی که دارند می‌توانند غیر فعال شوند، مواقعی مانند: اجرای یک اپلیکیشن و یا flash programming

سیستم عامل FreeRT از کرنل در برابر اجرای نامعتبر برنامه‌ها (تسک‌ها)ی کاربر جلوگیری می‌کند همچنین قابلیت تشخیص Stack Overflow را در سه ناحیه MPU براساس هر تسک (Thread) تشخیص می‌دهد. در این سیستم عامل واحد MPU به ندرت استفاده می‌شود و به خوبی پیاده‌سازی نشده است.

سیستم عامل Zephyr یک سیستم عامل Open Source است که برای دستگاهای با منابع محدود طراحی شده که مهمترین رسالتش انتعطاف پذیری، کارایی و امنیت بوده. امکاناتی برای حفاظت از حافظه و امنیت ارائه میده. اما به اشتراک گذاشتن حافظه بین هاThread ممکنه که موجب کاهش ایزوله‌سازی حافظه و آسیب‌پذیری نقطه‌ای شود.

[۱] Memory Protection Unit

۲ بخش الف تکامل به سمت IoTOS یا Evolution toward IoT OS

رسالت اصلی سیستم عامل TizenRT برای پروژه‌های محیط‌های کوچک می‌باشد. این سیستم عامل از نوع کرنل لینوکسی می‌باشد که بسیاری از معماری‌های نرم‌افزاری آن ارثبری شده از کرنل سیستم NuttX می‌باشد [۲].
دو تا از مهم‌ترین ویژگی‌هایی که سیستم عامل Tizen داره ارائه میده به موارد زیر میرسه:

۱. قابلیت fail-safe file system

۲. قابلیت Light Weight Database

این دو قابلیت تمام توابع مربوط به CRUD را بسیار مطمئن‌تر و آسان‌تر می‌کند.
تقریبا می‌توان به این نتیجه رسید که تمام وسایل خانگی هوشمند از سیستم عامل متن‌باز RT استفاده می‌کنند. مثل تصفیه کننده هوا، یخچال‌ها و کولرها. در کنار تمام این ویژگی‌ها، دستگاه‌های IoT باید UI خوبی برای تعامل کاربر با سخت‌افزار را داشته باشند. از قبیل صفحه نمایش لمسی (برای دستگاه‌های پوشیدنی) شناسایی فرمان‌های صوتی. سیستم عامل TizenRT نه تنها از user interface framework استفاده می‌کند بلکه دارای یک دستیار صوتی هوشمند به اسم Bixby می‌باشد که در کنفرانس توسعه دهندگان سامسونگ در سال ۲۰۱۸ [۳] معرفی شد.
با استفاده از این دستیار صوتی نه تنها می‌توان یک دیود LED را خاموش و روشن کرد بلکه می‌توان به آن دستور پخش یک موسیقی دلخواه را داد. دستگاه‌های Headless computer کامپیوترهایی بدون مانیتور، کیبورد و ماوس هستند. این سیستم کامپیوتری را می‌توان درون شبکه قرار داد. نیت اصلی این دستگاه‌ها کاهش هزینه‌های عملیاتی است.

۳ بخش ب تردها، تسک‌ها و باینری (نرم‌افزارها)

برای درک بهتر اینکه TizenRT چطوری کار میکنه و چطوری بخش level User رو به شکل مطمئن مدیریت میکنه، نیازمند این هستیم که بدانیم تردها، تسک‌ها و باینری‌ها چقدر در این سیستم عامل نسبت به سیستم عامل‌های دیگر متفاوت هستند که این سیستم انقدر مطمئن و پایدار تعریف شده است.

۱.۳ Thread

ک واحد برای زمانبندی است و چند Thread می‌تواند به صورت گروهی، انجام یک Task را بر عهده گیرد. (شکستن یک Task به هایSubtask مساوی و اختصاص هر یک از آنها به هایThread مختلف)

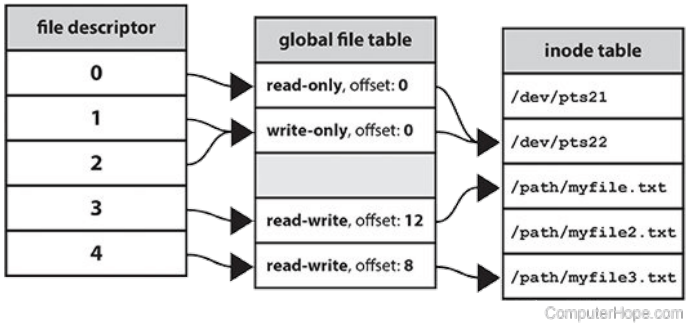
۲.۳ تسک

قسمتی از برنامه برای انجام یک کار مشخص و تخصصی می‌باشد. معمولا در یک برنامه بیشتر از یک Task در حال انجام می‌باشد [۴] مانند File Descriptor.

اگر یک Thread در یک فایل باینری کاربر باعث خطای حافظه شود، احتمالا کل فایل باینری متحمل خطا می‌شود زیرا همه هاThread فایل باینری را در یک حافظه مشترک به اشتراک می‌گذارند. (اشاره به خواندن و نوشتن در Shared Memory). به همین دلیل بایستی یک واحد برای جدایی خطا (Fault Isolation) و بازیابی سریع (Fast recovery) در فایل باینری وجود داشته باشد.

۳.۳ منظور از File Descriptor

در سیستم عامل کامپیوتری هر برنامه‌ای که کاربر می‌خواهد آن را اجرا کند یک عدد منحصر به فرد نامنفی به عنوان شناسه به آن برنامه اختصاص می‌یابد. این شناسه source Data را تعریف می‌کند و مشخص میکند که واحدهای مختلف مانند واحد حافظه چگونه و با چه شناسه‌ای می‌تواند به آن دسترسی داشته باشد. File Descriptor برای اولین بار در سیستم عامل Unix استفاده شد و سپس بعد از آن



شکل ۱: قسمتی از فعالیت مربوط به File Descriptor

سیستم عامل‌های مدرن مانند Linux، MacOS و حتی Windows و BSD از آن استفاده کردند. این عمل در سیستم عامل ویندوز به نام File handles می‌باشد.

مراجع

[۱] Samsung، S/W Platform for Digital Appliance: Part I. TizenRT، SDC۲۳ (۲۰۲۱).

[۲] NuttX-Gregory Nutt، Available: https://cwiki.apache، Online (۲۰۲۰).

[۳] S. Sahu، "TizenRT demo: Use bixby to control SmartThings-enabled devices"، Presented at the Samsung Develop. Conf. (SDC), San Francisco, CA, USA (۲۰۱۹).

[۴] Available: https://github.com/Samsung/TizenRT/، Online (۲۰۲۰).