



ارائه شده توسط:

سایت ترجمه فا

مرجع جدیدترین مقالات ترجمه شده

از نشریات معتبر

بررسی زمان واقعی سیستم عامل: RTOS

چکیده:

این مقاله در مورد بررسی منابع RTOS (زمان واقعی سیستم عامل) و سهم آن در جهان ایجاد شده است. RTOS به عنوان یک سیستم است که در آن صحت سیستم تنها به نتایج منطقی محاسبات بستگی ندارد بلکه به زمان تولید نتایج نیز بستگی دارد. انجام وظایف حیاتی بر اساس اولویت حداقل زمان انتظار است. به این دلیل اغلب چند تصور غلط در ارتباط با این مسئله به وجود آمده است و ما تلاش کرده ایم این مسائل را پاسخ دهیم. از آنجا که در 20 سال گذشته، RTOS تحت تکامل مداوم است و RTOS به توسعه بسیاری از محصولات تجاری منجر شده است. ما چند RTOS تجاری از دسته های مختلف برنامه های زمان واقعی انتخاب کرده و ویژگی های زمان واقعی آن ها را مورد بحث قرار داده ایم. مقایسه ی RTOS های تجاری ارائه شده است. ما با بحث در مورد نتایج حاصل از بررسی و مقایسه RTOS بر اساس پارامترهای عملکرد نتیجه گیری کرده ایم.

لغات کلیدی: RTAI، VxWorks

I. مقدمه

طی چند سال اخیر، افزایش قابل توجهی در تقاضای سیستم های تعبیه شده در امور انسانی وجود داشته است. در هر زمینه اعم امور خانه شخصی، کسب و کار، و فرایند اتوماسیون در صنایع، ارتباطات، سرگرمی و دفاع ساخته شده است. همانطور که می دانیم، سیستم های جاسازی شده ترکیبی از سخت افزار و نرم افزار و شاید دیگر قطعات مکانیکی طراحی شده برای انجام یک تابع [1] خاص هستند. عملکرد این سیستم بستگی به سیستم عامل استفاده شده دارد. بسیاری از این سیستم ها نیاز به RTOS دقیق برای انجام چنین وظیفه دارند.

اولین RTOS بیش از 20 سال پیش توسط خانواده PDP از ماشین آلات ساخته شد، که پس از آن روند تکامل ایجاد شد. زمان واقعی سیستم عامل (RTOS)، به عنوان نام نشان یک مهلت مرتبط با وظایف فراهم می کند و RTOS پایبند به این مهلت به عنوان یک مهلت از دست رفته می تواند باعث تحت تاثیر قرار گیری و اثر نامطلوب و

فاجعه بار شود. RTOS باید قطعی و پیشگیرانه باشد. RTOS موثر است و اجازه می دهد تا برنامه های زمان واقعی طراحی شود و به راحتی اجرای مورد نیاز گسترش یافته است.

اختصارات و مخفف ها :

IPC - ارتباط بین پردازنده

MPU - ریزپردازنده واحد

DEC - شرکت تجهیزات دیجیتال

LoC - خط از کد

یک سیستم عامل به طور کلی از دو قسمت تشکیل شده است: فضای هسته (حالت هسته) و فضای کاربر (حالت کاربر). این بخش اصلی هر سیستم عامل است و به عنوان یک پل ارتباطی بین برنامه ها و پردازش داده های واقعی در سطح سخت افزاری عمل می کند. هسته می تواند پایین ترین سطح لایه انتزاعی را برای ارائه منابع (به خصوص پردازنده ها و دستگاههای I/O) ایجاد کند.

انواع هسته در زیر مورد بحث است :

A. هسته یکنواخت:

- سیستم های یکنواخت موسوم به کد اسپاگتی یا بیگ مس هستند.
- این نوع هسته در اوایل برجسته بود.
- در اینجا، سیستم مجموعه ای از روش ها است.
- هر ماژول می نامد هر ماژول های دیگر.
- بدون پنهان کردن اطلاعات (به عنوان ماژول، بسته به کلاسی که در حال حاضر مورد استفاده قرار گرفته) با میلیون ها LOC و 16/1 اشکالات در LOC 1000 مورد سیستم های یکنواخت به احتمال زیاد شامل بسیاری از اشکالات. این نوع از سیستم عامل دارای مشکل دشوار به اشکال زدایی است. از دیدگاه قابلیت اطمینان در سطح بالا، یک هسته یکنواخت بدون ساختار است.

B. میکرو هسته:

- در اینجا، کد تا آنجا که ممکن از هسته به فضای "کاربر" حرکت می کند.
- بین ماژول و کاربر با استفاده از پیام ارتباط انجام می گیرد.
- گسترش یک هسته کوچک و پورت سیستم عامل به ساختار جدید آسان تر است.
- قابل اعتماد تر (کد کمتر در حالت کارکرد هسته) و امن تر است.

C. هسته خارجی:

- در اینجا هسته سخت افزار را از نرم افزار جدا می کند.
- هسته منابع فیزیکی را به برنامه اختصاص داده است. به ایت دلیل در مدل مفهومی بر اساس (فایل های سیستم، فضای آدرس مجازی، برنامه، سوکت) است.
- برنامه تصمیم می گیرد تا با این منابع چه کار کند. می تواند به یک OS برای شبیه سازی OS متعارف متصل شود.
- معمولا در آن تعامل سخت افزار قوی مورد نیاز است.

RTOS می تواند به عنوان "توانایی سیستم عامل برای ارائه یک سطح مورد نیاز از خدمات در یک زمان پاسخ محدود" تعریف شود. [2] RTOS ها 'عموما دو دسته هستند، یعنی، زمان واقعی سخت و زمان واقعی نرم و طبقه بندی آن ها به شرح زیر است:

- A. RTOS سخت: این نوع از RTOS به شدت به مهلت مرتبط با وظایف پایبند است. این سیستم می تواند هر گونه تاخیر را تحمل کند، در غیر این صورت سیستم را از بین خواهد برد. برای مثال، ضربان ساز قلب.
- B. RTOS نرم: در این نوع از RTOS، زمان از دست رفته یک مهلت قابل قبول است. این سیستم می تواند تاخیر را تحمل کند اما بسیار نامطلوب است. به عنوان مثال، بر روی خط پایگاه داده. علاوه بر این، آن ها با توجه به نوع دستگاه های سخت افزاری (به عنوان مثال 8 بیتی، 16 بیتی، 32 بیتی MPU) پشتیبانی می شوند.

II. ویژگی های RTOS:

RTOS باید به روشی طراحی شود که یک توازن بین حمایت از مجموعه ای از ویژگی ها برای توسعه و استقرار برنامه های کاربردی زمان واقعی و به خطر انداختن مهلت ایجاد شود. [3]

A. اولویت کار:

اولویت قابلیت تعریف برای شناسایی کار، نیاز به منابع و تخصیص آن برای ایجاد کنترل در RTOS دارد. مانند قابلیت اختصاص وظیفه منحصر به فرد با سطح اولویت مناسب. بنابراین، مهم است که RTOS به این ویژگی مجهز شده است.

B. مکانیسم مطمئنو کافی ارتباط بین کاری:

برای کارهای مختلف و برای برقراری و برای اطمینان از درستی داده ها در هر مکانیزم ارتباط بین کاری و هماهنگ سازی، مکانیسم قابل اعتماد و کافی مورد نیاز است.

دیگر ویژگی های RTOS در طول تجزیه و تحلیل عملکرد در ادامه این مقاله بحث شده است.

در اینجا RTOS از دسته بندی های مختلف انتخاب شده است:

- VxWorks : سازگار ترین RTOS

- Windows CE : موفق از نظر تجاری

- QNX Neutrino : متداول در سیستم های گره چند گانه.

- RTAI : رایج ترین RTOS

III. سوء تفاهم های مربوط به RTOS و شفاف سازی آن ها

- RTOS بسیار سریع است : این درست نیست.

RTOS باید رفتار قطعی از نظر مهلت داشته باشند اما این درست نیست که سرعت پردازش یک RTOS سریع

است. توانایی پاسخگویی RTOS به این معنا نیست که آن ها سریع هستند.

• همه RTOS ها شبیه هستند.

همانطور که قبلا مورد بحث قرار گرفت دو نوع RTOS (سخت و نرم) وجود دارد.

- RTOS مقدار قابل توجهی از بار CPU را استفاده می کند.
- این درست نیست. تنها 1-4٪ از CPU توسط RTOS مورد نیاز است. [4].
- هیچ علمی در طراحی سیستم RTOS وجود ندارد.
- ولی بیشتر علوم خوب از تلاش برای حل مسائل پیش روی با سیستم های مختلف کار گذاشته شده کارآمد بر گرفته شده است.
- RTOS همیشه انرژی زیادی مصرف می کند.
- مصرف انرژی هر سیستم به طور عمده وابسته به سخت افزار و ساخت آن است.

IV. محدودیت های RTOS:

- می تواند پر هزینه باشد.
- RTOS به طور کلی بسیار پیچیده است و مقدار غیر بدیهی از چرخه پردازنده را مصرف می کند.
- RTOS دارای چند وظیفه عدم زمانبندی با نوبت گردشی را پشتیبانی نمی کند.

V. RTOS زیر مقایسه:

- A. VxWorks :** VxWorks یک RTOS طراحی شده توسط سیستم رودخانه ای ، کالیفرنیا است. این RTOS تجاری بیشتر در جهان امروز ترجیح داده شده است. ماژول های خاصی دارد که توسط پروژه های بوئینگ 777 و مریخ نورد استفاده می شود. بنابراین بحث در مورد جزئیات این RTOS بسیار ضروری است. [5]
- B. Windows CE :** در حال حاضر رسماً به عنوان Windows جاسازی شده جمع و جور شناخته شده است. این O.S منبع بسته، اولین بار 16 سال پیش ، در سال 1996 منتشر شد. بسیاری از سیستم عامل های تعبیه شده بر روی هسته Windows CE O.S، از جمله Windows فون میکروسافت و بسیاری از دستگاه های صنعتی بر اساس آن بوده اند. [6]

C QNX Neutrino : یک یونیکس تجاری مانند است RTOS. در اصل در سال 1982 توسط شرکت کانادایی سیستم های کوانتومی نرم افزار QNX توسعه یافته است. یکی از اولین O.S های میکرو هسته تجاری موفق بود است و در انواع مختلف دستگاه ها از جمله خودرو و تلفن همراه استفاده می شود. [7]

D. RTAI : RTAI مخفف زمان واقعی نرم افزار رابط است. که به شما اجازه نوشتن برنامه های کاربردی با محدودیت زمان دقیق برای لینوکس را می دهد ، یک فرمت در زمان واقعی برای هسته لینوکس است. مورد استفاده برای کنترل ربات ها، سیستم اکتساب داده ها، کارخانه های تولیدی، و سایر ابزار حساس به زمان و ماشین آلات می باشد. افزایش قابلیت برنامه ریزی زمان واقعی سخت و شکل های هندسی اولیه برای برنامه های کاربردی برای آن استفاده شده است. [8]

VI. پارامترهای عملکرد تجزیه و تحلیل:

A. وقفه:

این یک وضعیت است که در آن دو یا چند فرآیند در حال انتظار برای به پایان رسیدن دیگر فرآیند ها می باشند. این حالت معمولاً در نبود منابع و یا در اجرای عقب تر ایجاد می شود. همانطور که می دانیم هر گونه تاخیر در نتیجه RTOS می تواند فاجعه بار باشد، بنابراین ما انتظار داریم RTOS ترجیحاً از وضعیت وقفه جلوگیری کند و یا به صورت موثر به آن رسیدگی کند.

شرایط وقفه:

- دو به دو ناسازگار
- انتظار مدور
- نگه داری و انتظار
- پیش بینی کمتر

این چهار شرط به عنوان شرط کافمن از اولین مقاله در سال 1971 توسط ادوارد جی کافمن بدست آمده است [9]

احتمال RTOS به شرط وقفه و به درجه پیش بینی بستگی دارد.

1) اجتناب از وقفه : بهترین روش برای جلوگیری از یکی از چهار شرط کافمن پیش از وقوع است. همچنین روش هایی مانند کسب اطلاعات اضافی در مورد فرآیند و تصمیم گیری یک فرایند می تواند توسط یک RTOS برای جلوگیری از وقفه انجام شود.

2) تغییرات در وقفه : اگر وقفه رخ می دهد، RTOS باید قادر به بهبود آن ، در اسرع وقت باشد. معمولا راه حل بهبود پایان روند است، که به صورت زیر اجرا می شود:

- حذف تمام فرایندهای وقفه.
- حذف یک فرایند در یک زمان تا چرخه وقفه حذف شود.

B. ردپای حافظه:

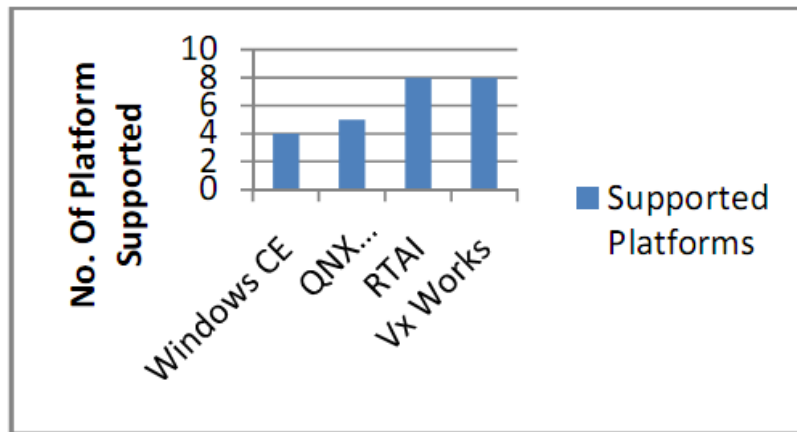
برآورد حافظه RAM و ROM از الزامات یک RTOS بر روی پلت فرم های جاسازی شده خاص است. کد موثر، فقط داده های خواندنی هسته، و هر کد زمان اجرا هر سه اندازه ROM مورد نیاز است ، از سوی دیگر یک مجموع ساختار داده ها و متغیرهای جهانی و برنامه های موقت است.

ارزش حافظه بر ساختار پلت فرم سخت افزار، کامپایلر تنظیمات (بهینه سازی) و از همه مهمتر تنظیمات سیستم عامل که شامل اندازه هسته و اندازه زمان اجرا بستگی دارد. معیارهای ردپا اغلب با توجه به یک راه حل RTOS مشخص شده است ، به خصوص در شرایطی که دستگاه ها بر روی تراشه حافظه دارند و امکان ارتباط با حافظه خارجی را محدود کرده اند یک عامل تصمیم مهم می باشد. [10]

ارزش ردپای پایین تر از RTOS ممکن است هزینه اختصاص داده شده برای سخت افزار ذخیره سازی یک پروژه را کاهش دهد.

C. قابلیت حمل / سازگاری:

اغلب، یک برنامه قابل اجرای سخت افزاری است که در اصل برای بزرگ تر شدن طراحی شده است ، الزامات محصول را افزایش می دهد. به این ترتیب RTOS با چنین قابلیت می تواند به ساختار پردازنده منتقل شود. بنابراین، انعطاف پذیری در انتخاب سخت افزار با توجه به نیاز پروژه را ارائه می دهد.



شکل 1. تعداد سیستم عامل های RTOS تحت بررسی حمایت

D. ابزارهای توسعه ارائه شده:

مجموعه ای کافی از ابزارهای توسعه از جمله دیباگر؛ کامپایلر و پیش فیلتر ممکن است در عملکرد کوتاه توسعه و اشکال زدایی زمان کمک کنند، و بهبود قابلیت اطمینان از برنامه نویسی را ایجاد کنند. RTOSs. تجاری معمولاً مجموعه ای کامل از ابزارها برای تجزیه و تحلیل و بهینه سازی رفتار RTOSs دارند، در حالی که منبع باز RTOSs ممکن است همان را نداشته باشد. به عنوان مثال. سیستم عامل Windows CE محبوب ترین ابزار توسعه پیشرفته و کارآمد (مایکروسافت ویژوال استودیو 2012) را فراهم می کند. ما برای این مقایسه در نظر گرفته ایم.

E. امنیت ارائه شده:

تا دهه گذشته، RTOS دارای ویژگی های معمولی بوده است، اما از چند سال گذشته، با توجه به حملات برنامه های مخرب بر روی سیستم های مهم و شبکه، تامین امنیت کامل انجام شده. در اینجا امنیت به حفاظت سیستم امنیت از دسترسی های غیر مجاز در داخل سیستم و همچنین در خارج از سیستم اشاره دارد. انتظار می رود آن ها به دنبال استانداردهای امنیت POSIX برای خدمت به برنامه های تعبیه شده موجود در RTOS مانند VxWorks و هسته (MG) باشند [11].

F. عملکرد زمان اجرا:

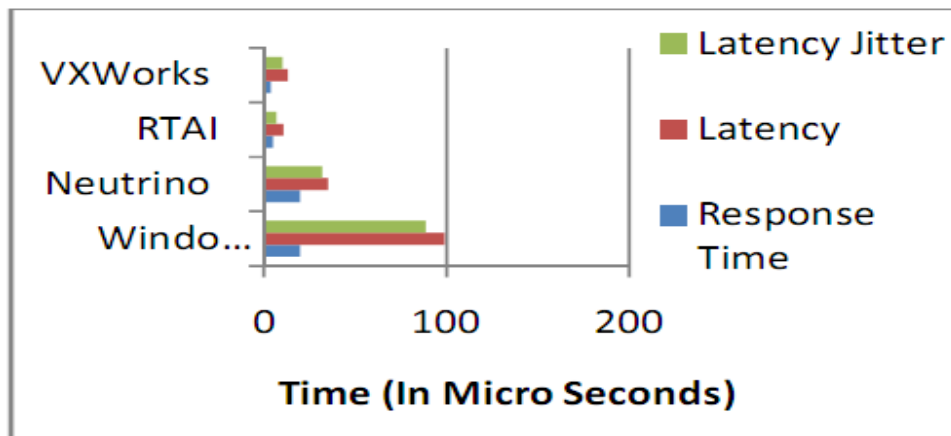
زمان اجرا عملکرد یک RTOS به طور کلی توسط حالت وقفه تحت تاثیر است. این نظر مفید است اگر ارزیابی عملکرد از برنامه در RTOS به نمونه اولیه جنبه های عملکرد حیاتی آن بر روی سخت افزار استاندارد داده شود [12].

1) زمان تاخیر: زمان تاخیر تجزیه و تحلیل خارجی RTOS تحت آزمون در رابطه با سخت افزار به عنوان یک جعبه سیاه است. تاخیر متشکل از اختلاف زمان بین فرستادن و لحظه ای که برنامه عمل می کند می باشد، کنترل وقفه همراه یک واکنش خارجی انجام می شود.

2) جیتیر: جیتیر اطلاعات غیر مستقیم بدست آمده از چندین شاخص وقفه می باشد که متشکل از یک تغییر تصادفی بین هر مقدار تاخیر است. در یک RTOS، اثر جیتیر زیاد است و با پروکتور در زمان کنترل موتور مرحله ای قابل تحلیل است.

3) بدترین حالت زمان پاسخ: بدترین حالت زمان پاسخ با استفاده از روش ارائه شده توسط ISA است که در بالا تجزیه و تحلیل فرکانس حداکثر وقفه است که توسط RTOS با قابلیت اطمینان به کار گرفته مورد بحث قرار گرفته به دست آمده است. بدترین زمان پاسخ مورد معکوس حداکثر فرکانس به دست آمده می باشد.

(آزمون پیشنهاد شده توسط A & CS شامل تنظیم یک سیستم است که نسخه سیگنال ورودی به طور مستقیم به یک پورت خروجی است، و اندازه گیری میزان پالس در ورودی ایجاد شد. از لحاظ تئوری، در حالی که سیستم پایدار است، تعداد پالس انباشته شده در هر دو پورت باید برابر باشد. پس از آن، فرکانس سیگنال ورودی باید به آهستگی افزایش یابد تا پالس ورودی و خروجی در شمارش اختلاف داشته باشند. در این لحظه، فرکانس باید کاهش یابد تا زمانی که فرکانس کار سیستم حداکثر مشخص شود.) [13]



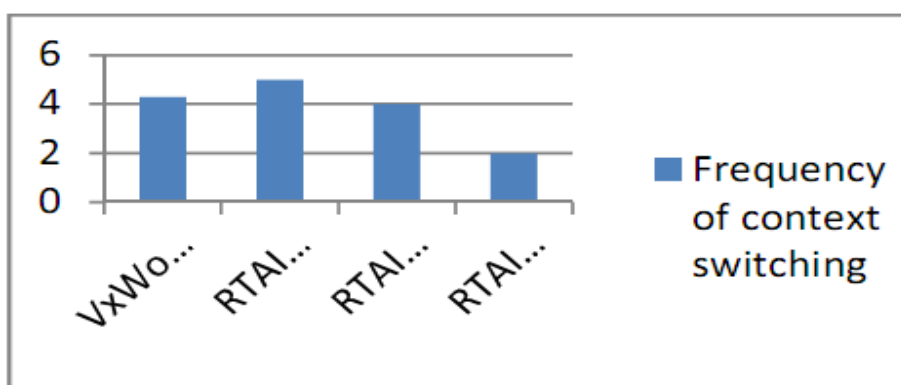
شکل 3. نمایش بدترین زمان پاسخ

A: زمان پاسخ (1 / حداکثر فرکانس پایدار)، B: زمان تاخیر، C: لرزش زمان تاخیر

4) الگوریتم های زمان بندی: الگوریتم های زمان بندی در O.S مورد استفاده می باشند، تأثیر زیادی بر عملکرد زمان اجرا دارند. مشکلات مربوط به چند وظیفه و اولویت توسط این الگوریتم پرداخته شده است. فرکانس سوئیچینگ زمینه نیز توسط این الگوریتم مشخص شده است.

الگوریتم زمان بندی پایه: اولویت پیشگیرانه به ترتیب وظایف اولویت های خود را انجام داده است. وظایف مورد نیاز دوره بالاتر باید اولویت بالاتری داشته باشند. وظایف محاسبات فشرده باید اولویت های پایین تر داشته باشند. VxWorks دارای گزینه ای برای فعال کردن حالت گرد رابین است، که در آن هر کار CPU برابر با نوبه خود می شود.

زمان بندی RTAI: FPPS (اولویت ثابت برنامه ریزی پیشگیرانه) در RTAI استفاده می شود، که در آن شغل با اولویت بالاتر برای اولین بار به پایان رسید. اولویت دوباره بر اساس محدودیت های زمانی مشخص شده است. برنامه ریزی پیشگیرانه از مشکل سوئیچینگ زمینه رنج می برد. برنامه ریزی اولویت ثابت با به تعویق افتاده (FPDS) به جای FPPS ارائه شده است. FDPS یک منطقه بین FPPS و FPNS (اولویت ثابت برنامه ریزی غیر انحصاری) است و مزایای هر دو تکنیک را ارائه می دهد. هر کار FPDS تنها بین زیر شغل امکان پذیر است. پیشگیری کمتر، تعویض زمینه کمتر را موجب می شود. [14]

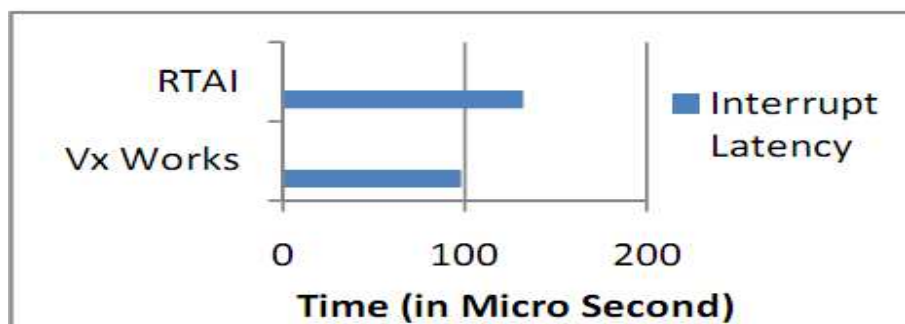


شکل 4. این شکل فرکانس سوئیچینگ زمینه تحت بررسی RTOS را نشان می دهد.

5) زمان تاخیر وقفه

تاخیر وقفه به عنوان مجموع تعریف وقفه زمانی است که هسته در انتظار پاسخ به یک وقفه است، صرفه جویی در زمینه وظایف، تعیین منبع وقفه، و فراخوانی کنترل وقفه را شامل می شود. برای یک وقفه خاص، زمان تاخیر نیز شامل زمان اجرای دیگر وقفه تو در تو است. از آنجا که سیستم ها قطع محور تعبیه شده اند، کم پوشیدگی وقفه توان عملیاتی سیستم را به شدت افزایش خواهد داد. (در این آزمایش، ما تایمر سخت افزار MPC8260 با یک دوره پیکربندی 50 MHz برای تولید یک وقفه تایمر هر 20 ثانیه است) [15]

Table Number	Calculation of Interrupt Latency	
	Vx Works	RTAI
Interrupt latency (μ S)	98(0.55)	132(1.2)

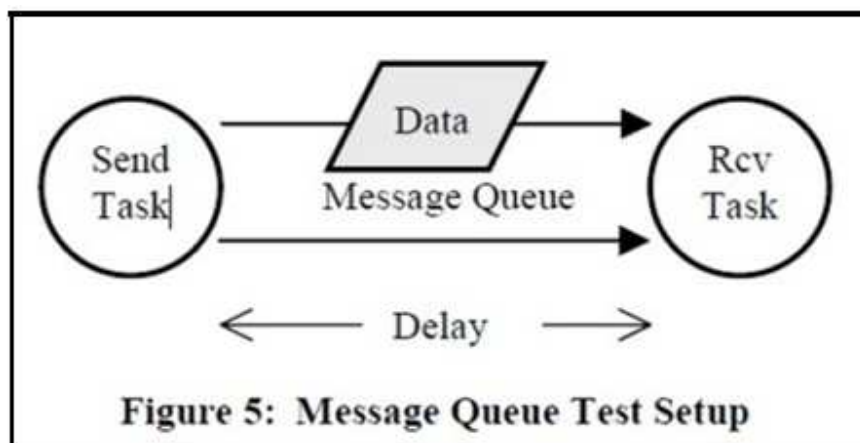


شکل 5. تاخیر وقفه RTAI و VxWorks را نشان می دهد

این تعجب آور نیست که VxWorks تاخیر وقفه بسیار پایین تر (35٪) از RTAI داشته باشد. لینوکس قدیمی با داشتن پوشیدگی وقفه بالا شناخته شده است. به نظر می رسد که حتی اگر RTAI با قابلیت زمان واقعی اضافه شود ، هنوز هم نمایش برخی از رفتار زمان واقعی است.

6) ارتباط بین فرآیندی

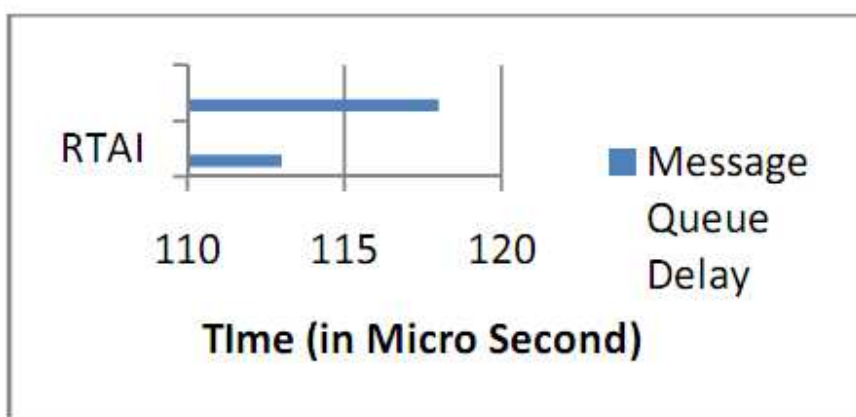
برنامه های مدرن زمان واقعی به عنوان مجموعه ای مستقل، از وظایف تعاونی ساخته شده است. همراه با سمافور با سرعت بالا، VxWorks و RTAI نیز به عنوان مکانیزم هماهنگ سازی سطح بالاتر ارائه اجازه می دهند تا همکاری وظایف برای برقراری ارتباط با یکدیگر انجام شود. به دلیل پیچیدگی پیاده سازی، استفاده از این سرویس بیشترین میزان تاخیر را تحمیل می کند و در نتیجه یک کلید متریک برای مطالعه سیستم عامل است. [16]



شکل 6. این شکل، شرایط آزمایش صف بندی پیام را نشان می دهد

این آزمایش با ایجاد صف بندی پیام انجام می شود. بعد، پیام تابع فراخوانی می شود. دریافت سیستم قدرت پاسخ وظیفه دریافت و آن را در حالت انتظار (از صف بندی پیام خالی است) قرار می دهد. در حالی که وظیفه دریافت برای پیام بود، ما بر اساس آن ها یک وظیفه ارسال پیام از طریق صف بندی پیام ایجاد می کنیم. زمان بین وظیفه ارسال با فراخوانی تابع ارسال پیام و وظیفه دریافت با اطلاع رسانی پیام در جدول زیر آورده شده است [17]:

Table Number	Calculation of Message Queue Delay	
Message Queue Delay (μ S)	VxWorks	RT AI
	118 (0.9)	113(1.8)



شکل 7. این شکل صف بندی تاخیر پیام در VxWorks و RTAI را نشان می دهد.

VII. نتیجه گیری:

برنامه های کاربردی زمان واقعی با توجه به پیچیدگی در این روزها به سیستمی محبوب تبدیل شده اند. برای مقابله با این پیچیدگی ها، توسعه دهندگان، نرم افزار زمان واقعی داده را ساخته اند. تعداد بسیار زیادی از RTOS موجود در بازار وجود دارد. به طوری که طراحی سیستم های کارآمد جاسازی شده از لحاظ هزینه را فراهم می کنند، مصرف برق، قابلیت اطمینان، سرعت و غیره یک مسئله و دشواری در رتبه بندی RTOS است چرا که بسیاری از انتخاب های خوب در بازار موجود است. توسعه دهنده می تواند هم RTOS تجاری (با استفاده از 44٪ توسعه دهندگان) یا RTOS موتورلا (20 درصد) و یا RTOS داخلی توسعه یافته (17 درصد) را انتخاب نماید. از مقایسه و مطالعه RTOS تجاری انتخاب شده، ما می توانیم نتیجه بگیریم که

- RTAI و VxWorks طیف گسترده ای از سیستم عامل های ارائه شده را پشتیبانی می کنند.

- RTAI، منبع باز RTOS برای برنامه های کاربردی کوچک، مانند رباتیک و دستگاه های پزشکی مناسب است.

- در حالی که، VxWorks غالباً در برنامه های کاربردی با عملکرد بسیار پیچیده و بالا است.
- هنگام وقفه انتخاب VxWorks بهتر است و سریع تر از RTAI است. برنامه های کاربردی در حال گرفتن تعداد وقفه مداوم در VxWorks باید ترجیح داده شوند. افزایش وقفه ها در تعداد ورودی به RTOS همزمان استفاده می شود.
- از نظر ارسال / دریافت پیام ، RTAI یک عملکرد بهتر از VxWorks ارائه می دهد. همانطور که قبلاً ذکر شد، این ارقام ممکن است تا حد زیادی وابسته به پیاده سازی متفاوت IPC (IPC) را می توان با استفاده از حافظه مشترک اجرا کرد) باشند.



این مقاله، از سری مقالات ترجمه شده رایگان سایت ترجمه فا میباشد که با فرمت PDF در اختیار شما عزیزان قرار گرفته است. در صورت تمایل میتوانید با کلیک بر روی دکمه های زیر از سایر مقالات نیز استفاده نمایید:

لیست مقالات ترجمه شده ✓

لیست مقالات ترجمه شده رایگان ✓

لیست جدیدترین مقالات انگلیسی ISI ✓

سایت ترجمه فا ؛ مرجع جدیدترین مقالات ترجمه شده از نشریات معتبر خارجی