

جزوه دوم سیستم های کنترل بلادرنگ

سیستم های کنترل بلادرنگ: سیستم عامل بلادرنگ نوعی سیستم عامل است که در آن، زمان، پارامتر کلیدی است. برای مثال در سیستم های کنترل فرایند، رایانه های بی درنگ باید داده های فرآیند تولید را جمع آوری کرده و به کمک آن ماشینهای داخل کارخانه را کنترل کنند. سیستم هایی که آزمایش های علمی را کنترل می کنند، سیستم های تصویربرداری پزشکی، سیستم های کنترل صنعتی، کنترلرهای لوازم خانگی و سیستم های جنگ افزاری، سیستم های بی درنگ هستند.

سیستم بلادرنگ، یک سیستم نرم افزاری است که درون یک دستگاه قرار گرفته است. ما این دستگاه را مستقیماً نمی توانیم ببینیم چون از این دستگاه باید به صورت توکار، در جاهای مختلف استفاده کرد، یعنی مثلاً در اتاق خلبان درون هواپیما، ما نمی توانیم این دستگاه را مستقیماً ببینیم بلکه این دستگاه درون تجهیزات موجود در اتاق خلبان کار گذاشته می شود و با تمام تجهیزات موجود، در ارتباط است.

مشخصات سیستم عامل های بلادرنگ

- قطعی بودن
- پاسخدهی
- کنترل کاربر
- قابلیت اطمینان
- نرمش با خطا

انواع سیستم های بلادرنگ

سیستم های Real-time به دو دسته تقسیم می شوند :

بلادرنگ سخت (Hard Real-time)

بلادرنگ نرم (Soft Real-time)

بلادرنگ سخت سیستمی است که در یک مهلت زمانی یا پاسخ میدهد یا هیچ. مانند سیستم مانیتورینگ بخش I.C.U یک بیمارستان، سیستم های کنترل صنعتی، موشکها، ماهواره ها...

سیستم بلادرنگ نرم سیستمی است که در بعضی از مواقع، آماده نشدن پاسخ در مهلت زمانی تعیین شده قابل تحمل است مانند مالتی مدیا

کنترلهای بی درنگ

در بحث بیدرنگ دو مفهوم اساسی فرآیند و فرآیند صنعتی تعریف میشود:

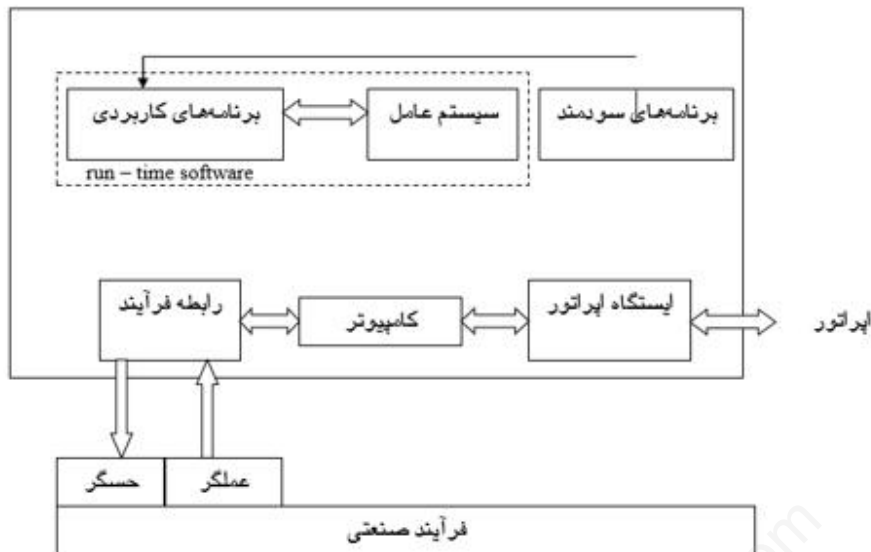
• فرآیند: عبارتست از کلیه فعالیتهای یک سیستم که بر یکدیگر اثر گذاشته و با آن ماده، انرژی یا اطلاعات را تبدیل، منتقل یا ذخیره میکند.

• فرآیند صنعتی: فرآیندی است که کمیتهای فیزیکی آن را بتوان با وسایل فنی بدست آورد و تحت تأثیر قرار داد. فرآیند صنعتی را میتوان به فرآیندهای تولیدی، توزیعی، ذخیره ای طبقه بندی کرد.

برای نگهداری فرآیند در یک حالت دلخواه باید آن را مطابق با قوانین معینی کنترل کرد این قوانین بوسیله سیستم کنترلی ایجاد میشود که ممکن است اپراتوری (دستی)، خودکار یا ترکیبی از این دو باشد. دلیل استفاده از سیستمهای کنترل کامپیوتری این است که بسیار قابل اعتماد، دقیق و ارزان هستند.

عناصر سخت افزاری یک سیستم بیدرنگ

عناصر سخت افزاری يك سیستم بیدرنگ



جدول مراحل اصلی در تکنولوژی سیستم‌های بی‌درنگ

سخت افزار	نرم افزار	روش
کامپیوترهای متمرکز بزرگ / کابل کشی ستاره‌ای شکل پایانه‌های ابتدایی / رابط فرآیند ابتدایی	بدون سیستم عامل	روش‌های دلخواه
مینی کامپیوترها - سیستم‌های چند کامپیوتری - ساختارهای گذرگاهی پایانه‌های گرافیکی	برنامه‌های اجرایی چند تکلیفی زبان‌های سطح بالا	برنامه سازی ساخت یافته
ریز کامپیوترها - شبکه‌های توزیع یافته - پایانه‌های هوشمند رابط فرآیند هوشمند	رابط خوشکار به برنامه‌های سیستمی بسته برنامه‌های کاربردی	طراحی ساخت یافته ابزار CASE تحلیل ساخت یافته
1990	سیستم‌های عامل هوشمند زبان‌های بی‌درنگ توزیع یافته	ابزار ساخت و تولید نرم افزار برای چرخه کامل زندگی

وظایف سیستم های بلادرنگ

سیستم عامل دارای وظایف زیر است :

- مدیریت پردازنده
- مدیریت حافظه
- مدیریت دستگاهها (ورودی و خروجی)
- مدیریت حافظه جانبی
- اینترفیس برنامه های کاربردی
- رابط کاربر

وظایف شش گانه فوق، هسته عملیات در اکثر سیستم های عامل است.

مدیریت پردازنده

مدیریت پردازنده دو وظیفه مهم اولیه زیر را دارد:
ایجاد اطمینان که هر پردازنده و یا برنامه به میزان مورد نیاز پردازنده را برای تحقق عملیات خود، اختیار خواهد کرد.
استفاده از بیشترین سیکل های پردازنده برای انجام عملیات

مدیریت حافظه و فضای ذخیره سازی

سیستم عامل در رابطه با مدیریت حافظه دو عملیات اساسی را انجام خواهد داد:
• هر پردازنده بمنظور اجراء می بایست دارای حافظه مورد نیاز و اختصاصی خود باشد.
• از انواع متفاوتی حافظه در سیستم استفاده تا هر پردازنده قادر به اجراء با بالاترین سطح کارائی باشد.

مدیریت دستگاهها

دستیابی سیستم عامل به سخت افزارهای موجود از طریق برنامه های خاصی با نام "درایور" انجام می گیرد. درایور مسئولیت ترجمه بین سیگنال های الکترونیکی زیر سیستم های سخت افزاری و زبانهای برنامه نویسی سطح بالا و سیستم عامل و برنامه های کاربردی را برعهده خواهد داشت. مثلا درایورها اطلاعاتی را که سیستم عامل بصورت یک فایل تعریف و در نظر می گیرد را اخذ و آنها را به مجموعه ای از بیت ها برای ذخیره سازی بر روی حافظه های جانبی و یا مجموعه ای از پالس ها برای ارسال بر روی چاپگر، ترجمه خواهند کرد.

در حالت کلی برای سیستم های بلادرنگ سه مرحله مورد بررسی است:

- Scheduling زمانبندی
- Programming برنامه نویسی
- Verification بازمیابی

زمانبندی

در یک سیستم بلادرنگ رویدادهایی رخ می دهد؛ برنامه به تعدادی پروسس تقسیم می شود و هر پروسس برای پاسخگویی به یک نوع رویداد است. رویدادهای قابل پاسخگویی در سیستم بلادرنگ به دو بخش زیر تقسیم میشود :

- وقایع متناوب: Periodic
- غیر متناوب: Aperiodic

رویداد متناوب با دوره تناوب مشخص تکرار می شوند. رویداد غیر متناوب به صورت تصادفی رخ می دهند (زمان رخ داد مشخصی ندارند). از آنجا که پردازش مربوط به هر رویداد بخشی از زمان CPU را اشغال می کند، ممکن است پاسخ کلیه وقایع در مهلت مشخص امکان پذیر نباشد.

برنامه نویسی

تمامی سیستم های بلادرنگ دارای یک نرم افزار بلادرنگ هستند و این نرم افزار به وسیله یک زبان برنامه نویسی خاص تهیه می شود، در واقع یکی از راه های ایمن سازی سیستم ها، انتخاب زبان برنامه نویسی مناسب کارا است. عموماً از ویژگی های این زبان ها شامل سادگی، خوانایی، بدون ابهام بودن، ایمن بودن، مستقل از زمان بودن و قابل حمل بودن است. همانطور که گفته شد این زبان ها بایستی بتوانند بلادرنگ عمل کنند و در نتیجه هر زبانی برای این کار مناسب نیست و در اینجا نمونه هایی از زبان های برنامه سازی بلادرنگ که از ابتدا تا کنون مورد استفاده قرار گرفته و می گیرد را مشاهده می کنید: Ada-Real-time, Erlang, Esterel, Real-time Euclid, C, Java, time, Posix, Lustre, Statecharts, SAFL, Chill, Modula-2, Mesa, FRP, RTL/2, Jovial, Coral 66 و ... است.

بازبینی

برای بازبینی سیستم های بلادرنگ از ابزاری به نام Uppaal استفاده می شود که ابزاری برای کنترل کردن سیستم بلادرنگ است و سیستم بلادرنگ را به صورت مدل نشان می دهد. در شکل یک محیط کار این ابزار مشخص شده است.

نتیجه گیری

در برخی کاربردها (مثل کنترل صنعتی) در کامپیوترها از سیستم عامل استفاده نمی شود. از آنجا که در سیستم های کنترل صنعتی برنامه می بایست در اسرع وقت در مقابل یک اتفاق، از خود عکس العمل نشان دهد، وجود واسطه سیستم عامل باعث کند شدن مراحل می گردد.

در اینجا پس از ارائه یک تعریف کلی از سیستم های بلادرنگ و کاربردهایشان، به بررسی الزام بکارگیری سیستم عامل بلادرنگ در چنین سیستم هایی می پردازیم. سپس ضمن بیان الزامات سیستم های عامل بلادرنگ و ویژگی های عمومی آنها، به بررسی پیش نیازهای یک سیستم عامل بلادرنگ به منظور تطابق با استاندارد POSIX 1003.1b پرداخته، و در پایان با معرفی چند سیستم عامل بلادرنگ رایج و ذکر چند نمونه از کاربرد آنها، به صحبت خاتمه می دهیم.

ارزیابی و مقایسه این سیستم عامل ها، بستگی به مدیریت حافظه و زمانبندی در آنها دارد. سیستم بلادرنگ، نوعی سیستم کامپیوتری که **صحت آن نه تنها به درستی منطق خروجی ها، بلکه همچنین به زمانی که طی آن نتایج تولید شده است، بستگی دارد.**

بنابراین تفاوت عمده یک سیستم بلادرنگ و غیر بلادرنگ این است که در یک سیستم بلادرنگ، مهلت زمانی یک وظیفه یا به عبارتی محدودیت زمان پاسخگویی به وقایع، دارای اهمیت زیادی است. بطوری اگر مهلت زمانی یک وظیفه رعایت نشود، نه تنها کیفیت عملکرد سیستم کاهش می یابد بلکه ممکن است منجر به لطمه های ناخواسته یا خطاهای غیر قابل برگشت به سیستم شود.

به عنوان مثال در یک نیروگاه هسته ای برخی کمیت ها باید همواره تحت کنترل دقیق باشند. مثلاً در یک راکتور هسته ای بر اثر پرتاب نوترون ها به اتم ها، نوترون های جدیدی آزاد می شوند و نوترون های آزاد شده نیز به اتم های دیگر برخورد کرده و نوترون های جدید آزاد می کنند و بهمین ترتیب. اگر تعداد نوترون های آزاد شده از یک حدی بیشتر شود، انفجار نوترونی رخ می دهد. پس غلظت نوترون ها همواره باید تحت کنترل دقیق باشد. حتی اگر یک ثانیه پس از انفجار، پاسخ مناسب داده شود، هیچ ارزشی ندارد.

به عنوان نمونه دیگری از یک سیستم بلادرنگ می توان سیستم مانیتورینگ بخش I.C.U بیمارستان را نام برد. یک تصور اشتباهی که درباره یک سیستم بلادرنگ وجود دارد این است که فکر می کنیم یک سیستم بلادرنگ، سیستمی با سرعت بالا است. در صورتیکه یک سیستم بلادرنگ الزماً یک سیستم با سرعت بالا نیست. یک سیستم می تواند خیلی کند کار کند، ولی در پاسخ دادن به رخدادها سرعت مناسبی داشته باشد.

مجدداً متذکر میشود که یک سیستم بلادرنگ باید مهلت زمانی وظایف را رعایت کند. این مهلت زمانی ممکن است 1 نانو ثانیه تا ساعت ها و روزها باشد.

در گذشته، سیستم های کامپیوتری بلادرنگ تنها برای کاربرد های خاص طراحی می شدند، اما در حال حاضر سیستم های جاسازی شده قابل برنامه نویسی و پیکربندی، عمومیت یافته اند و در زمینه های مختلف بکار گرفته شده اند.

به عنوان نمونه ای از کاربردهای سیستم های بلادرنگ می توان از استفاده آنها در سیستم های کنترل نیروگاه ها، کنترل دستگاه های صنعتی، کنترل ترافیک هوایی، کنترل تجهیزات نظامی و موشک ها، کنترل نسبت مخلوط سوخت و هوا در کاربراتور خودرو، سیستم های تعویض خطوط راه آهن، سیستم های مخابراتی، اجاق میکروویو و دستگاه های خودپرداز نام برد.

با توجه به کاربردهای سیستم های بلادرنگ و درجه حساسیت و فوریت واکنش به رخدادها خارجی، این سیستم ها به سه دسته-ی سیستم بلادرنگ سخت، سیستم بلادرنگ میان و سیستم بلادرنگ نرم تقسیم می شوند.

سیستم بلادرنگ سخت، سیستمی که برآورده نشدن محدودیت های زمانی در آن، باعث خراب شدن سیستم می شود. در چنین سیستمی ابدأ نباید هیچ رخدادی از دست برود. مانند سیستم های کنترل شرایط بحرانی در نیروگاه های هسته ای

در این سیستم های بلادرنگ میان نیز باید قبل از پایان یافتن مهلت زمانی به رخدادهای سیستم، پاسخ داده شود. پاسخ بعد از مهلت زمانی، غیر ممکن یا بی فایده است. ولی پاسخ ندادن باعث خرابی های جدی نمی شود. مانند سیستم های پیش بینی وضع آب و هوا

سیستم بلادرنگ نرم، سیستمی که کیفیت عملکردش، بخاطر برآورده نشدن محدودیت های زمان پاسخگویی کاهش می یابد، اما باعث خراب شدن سیستم نمی شود. مانند سیستم های مخابراتی

در سیستم های بلادرنگ بسته به سطح پیچیدگی عملکرد سیستم ممکن است از یک سیستم عامل بهره بگیریم. سیستم عامل غالباً در مواردی استفاده می شود که پیچیدگی عملکرد سیستم، یا نیاز آن به پروتکل های ارتباطی، وجود آن را اجباری کند، وگرنه در سیستم های صنعتی تاکید بر سادگی، ارزان تمام شدن و سریع بودن است.

به عنوان مثال، یک سیستم کنترل ماشین لباسشویی یا تلفن عمومی نیازی به سیستم عامل ندارد، اما یک خودپرداز به خاطر نیاز به ارتباط TCP/IP نیازمند استفاده از سیستم عامل است.

در کل یک سیستم بلادرنگ جهت برآورده ساختن محدودیت های زمانی اجرای برنامه و همچنین برای پشتیبانی از پروتکل های ارتباطی، الویت ها، وقفه ها، زمان سنج ها، ارتباط بین وظایف، همگام سازی، چند پردازشی و مدیریت حافظه، علاوه بر بکارگیری یک سخت افزار قوی، از یک سیستم عامل بلادرنگ بهره می گیرد.

نکته مهمی که در اینجا نباید فراموش شود این است که بکارگیری یک سیستم عامل بلادرنگ تنها بخشی (یک بلوک) از ایجاد یک سیستم بلادرنگ است. بدین ترتیب اگر در طراحی یا کدهای نوشته شده نکات یک سیستم بلادرنگ رعایت نشده باشد، سیستم عامل تاثیر چندانی در برآورده ساختن محدودیت های زمانی اجرا نخواهد داشت.

به عبارتی دیگر، نحوه برنامه نویسی، زبان برنامه نویسی، سطح برنامه نویسی، نحوه مدیریت پورت های I/O و خیلی مسائل دیگر در عملکرد سیستم تاثیر دارند.

به طور کلی به سیستم عاملی، سیستم عامل بلادرنگ گفته می شود که بتواند الزامات یک سیستم بلادرنگ را برآورده نماید. این الزامات به این شرح است:

قطعی بودن:

سیستم عاملی «قطعی» است که بتواند عملیات خود را در زمان های ثابت یا فواصل زمانی از پیش تعیین شده انجام دهد. علاوه بر آن سیستم عامل بلادرنگ باید قادر باشد با سرعت بیشتری نسبت به یک سیستم عامل معمولی، به یک وقفه با الویت بالا، شروع به ارائه خدمت نماید.

پاسخ دهی:

یک سیستم عامل باید قادر باشد پس از تصدیق یک وقفه، بسته به حساسیت و فوریت واکنش به رخداد، در زمان مناسب، خدمت به وقفه را انجام دهد.

کنترل کاربر:

در یک سیستم بلادرنگ لازم است به کاربر اجازه کنترل دقیق بر نحوه عملکرد سیستم داده شود. در یک سیستم بلادرنگ لازم است کاربر بتواند اولویت های وظایف را تعیین کند.

قابلیت اطمینان:

قابلیت اطمینان در سیستم های بلادرنگ بسیار مهم تر از سیستم های عادی است.

نرمش با خطا:

یک سیستم عامل بلادرنگ باید قادر باشد در صورت خرابی سیستم، تا حد ممکن قابلیت ها و داده هایش را حفظ نماید؛ همچنین در مواردی که ارضای تمام مهلت های زمانی غیر ممکن است، مهلت های زمانی وظیفه های بسیار حساس و با الویت بالاتر را برآورده نماید.

استاندارد POSIX 1003.1b، ضوابط پذیرفته شده ای برای سازگاری سرویس های RTOSها فراهم می آورد و به برنامه نویسان برنامه های کاربردی امکان نوشتن برنامه های قابل انتقال و اجرا روی انواع سیستم های عامل مبنی بر Unix را می دهد.

به منظور تطابق و سازگاری با استاندارد POSIX، سیستم عامل باید دارای ویژگی هایی به این شرح باشد:

چند وظیفه ای و قابلیت قبضه کردن:

به منظور پشتیبانی از چندین وظیفه گوناگون در کاربردهای بلادرنگ، RTOS باید دارای قابلیت چند وظیفه ای باشد و همچنین (زمانبند) بتواند هر وظیفه ای را در سیستم قبضه کند و منبع را در اختیار وظیفه دارای الویت بالاتر قرار دهد.

یک RTOS همچنین به منظور داشتن سطوح الویت متعدد، باید دارای سطوح متعدد وقفه ها باشد.

تشخیص مهلت زمانی بطور پویا:

به منظور دست یافتن به قابلیت قبضه کردن، RTOS باید بتواند به صورت پویا، وظیفه با زود ترین مهلت زمانی را تشخیص دهد. برای مدیریت مهلت های زمانی، ممکن است اطلاعات مهلت زمانی به سطوح الویت تبدیل شوند و برای تخصیص منابع مورد استفاده قرار گیرند.

همگام سازی قابل پیش بینی:

برای اینکه نخ های متعدد بتوانند بطور منظم و در زمان مناسب با یکدیگر ارتباط برقرار کنند، مکانیزم هایی برای ارتباط بین وظایف و همگام سازی با قابلیت پیش بینی مورد نیاز هستند. قابلیت پیش بینی پذیری نیز توسط حفظ یکپارچگی و صحت داده ها و، رعایت ترتیب اجرا به موقع وظایف امکان پذیر است. یکی از روش های حفظ یکپارچگی داده ها، قابلیت قفل کردن منابع می باشد.

سطوح الویت کافی:

در موقع استفاده از زمانبندی وظیفه الویت دار، RTOS باید برای پیاده سازی موثر دارای تعداد سطوح الویت کافی باشد. به عنوان مثال، جهت رسیدگی کردن مشکل الویت معکوس به سطوح الویت کافی نیاز داریم.

الویت معکوس که هنگامی رخ می دهد که وظیفه با الویت بالا منتظر وظیفه با الویت پایین تر برای آزاد کردن یک منبع مورد نیاز است، ولی به نوبه خود وظیفه با الویت پایین تر بخاطر وظیفه ای با الویت متوسط در انتظار است.

تاخیر های از پیش تعریف شده:

در یک سیستم عامل بلادرنگ، تاخیرهای تعویض وظیفه، اجرا اولین دستور گرداننده-ی وقفه و بازگشت از آن باید از قبل تعریف شده و قابل پیش بینی باشند.

حالا به معرف چند سیستم عامل بلادرنگ بیشتر مورد استفاده قرار گرفته اند، می پردازیم.

Windows CE

نسخه ای از سیستم عامل Windows شرکت Microsoft، برای کامپیوتر های کوچک و سیستم های جاسازی شده 32 بیتی سیار می باشد.

واضح است که Windows CE دارای هسته ای متفاوت از Windows های رومیزی می باشد.

این سیستم عامل روی پردازنده های Intel x86 پشتیبانی می شود و سازگار با پردازنده های ARM، MIPS، Hitachi SuperH و می باشد.

این سیستم عامل دارای 256 سطح الویت است.

هسته Windows CE قادر است در فضای حافظه کمتر از 1 مگابایت اجرا شود. بنابراین در دستگاه هایی که فاقد فضای ذخیره سازی هستند و همچنین به کاربر امکان هیچگونه توسعه ای را نمی دهد، در ROM دستگاه ذخیره می شود.

VxWorks

سیستم عامل بلادرنگی بر پایه Unix است که توسط شرکت Wind River Systems ایجاد و به فروش می رسد و همانند بسیاری از سیستم های عامل بلادرنگ دارای یک هسته چند وظیفه ای با قابلیت قبضه کردن و پاسخگویی سریع وقفه، امکانات ارتباطات بین فرایندی و همگام سازی، و یک سیستم پرونده می باشد.

VxWorks گسترده ترین استفاده به عنوان یک RTOS در صنعت سیستم های جاسازی شده را داشته و دارای یک محیط گرافیکی پیشرفته است.

این سیستم عامل دارای بیش از 1800 واسط برنامه سازی کاربردی است و روی سخت افزاری رایج قابل استفاده می باشد.

خصایص مشخص اصلی VxWorks شامل مدیریت حافظه منطبق با استاندارد POSIX، امکانات چند پردازنده ای، پوسته ای برای واسط کاربر، قابلیت اشکال زدائی در سطح نمادین و منبع، و نظارت بر عملکرد سیستمی باشد.

ریز هسته این سیستم عامل از 256 سطح الویت پشتیبانی می کند.

برخی سیستم های مخابراتی ساخت ایران از سیستم عامل VxWorks استفاده می کنند.

از برجسته تری کاربرد های این سیستم عامل می توان استفاده آن در کاوشگر مریخ را نام برد.

LynxOS

سیستم عامل بلادرنگی بر پایه Unix که توسط شرکت LynxWorks ارائه می شود.

این سیستم عامل منطبق با استاندارد POSIX و سازگار با Linux باشد و دارای ویژگی چند نخی است و برای کاربرد های بلادرنگ پیچیده ای که نیاز به پاسخگویی های سریع و قطعی دارند، طراحی شده است.

این سیستم عامل اساساً در سیستم های جاسازی شده بلادرنگ، در کاربرد هایی مانند ارتباطات فضایی و فضاوردی، سیستم های نظامی، کنترل فرایند تولید و مخابرات استفاده می شود.

این سیستم عامل دارای 512 سطح الویت می باشد.

VRTX

سیستم عامل بلادرنگ و چندوظیفه ای است که در حال حاضر توسط شرکت Mentor Graphics پشتیبانی و ارائه می شود و برای سیستم های جاسازی شده معمولی و سیستم های با معماری یک سیستم کامل روی یک تراشه (SoC) مناسب است.

VRTX بر روی سخت افزاری ARM، MIPS، PowerPC و دیگر میکروپروسسورها با معماری RISC قابل اجرا می باشد. این سیستم عامل در تلسکوپ فضایی هابل در حال استفاده است و رقیب سرسختی برای سیستم عامل VxWorks شناخته می شود.

QNX

سیستم عامل بلادرنگ تجاری بر پایه Unix و منطبق با استاندارد POSIX می باشد که اصولاً در بازار سیستم های جاسازی شده مورد استفاده قرار می گیرد.

QNX به عنوان یک سیستم عامل مبتنی بر معماری ریز هسته، همانند اغلب سیستم عامل ها در شکل تعدادی وظایف کوچک (خدمتگذارها) به اجرا در می آید.

رویکرد ریز هسته موجب انعطاف می شود و کاربر را قادر می سازد که بدون نیاز به تغییر دادن خود سیستم عامل، هر عملکردی را از کار بیاندازد. از طرف دیگر بخاطر معماری ریز هسته ایش، یک سیستم عامل توزیعی نیز هست.

این سیستم عامل بسیار کوچک است بر روی یک دیسک جا می گیرد و به عنوان سیستمی بسیار سریع و کاملاً منصف مطرح شده است.

QNX (2001) تقریباً بر روی اکثر پردازنده های جدیدی که در بازار سیستم های جاسازی شده مورد استفاده قرار می گیرد، قابل اجرا است.

این سیستم عامل دارای نسخه غیر تجاری به نام QNX NC بوده است که از سال 2003 توسعه و پشتیبانی از آن متوقف شده است.

RTLinux

سیستم عامل بلادرنگ RTLinux به عنوان توسعه ای از Linux، در ابتدا توسط V. Yodaiken ایجاد شد.

این سیستم عامل در دو نسخه رایگان (RTLinux Free) و تجاری (RTLinux Pro) موجود است.

RTLinux از طریق کنترل وقفه بین سخت افزار و سیستم عامل از عملیات بلادرنگ سخت پشتیبانی می کند.

وقفه هایی که نیاز به پردازش قطعی دارند توسط هسته بلادرنگ پردازش می شوند، در این حین بقیه وقفه ها به سوی سیستم عامل غیر بلادرنگ فرستاده می شوند.

سیستم عامل (Linux) به عنوان یک نخ با الویت پایین از هسته RTLinux اجرا می شود.

در بسیاری از کارهای تحقیقاتی در ایران و جهان از نسخه رایگان این سیستم عامل استفاده می شود.

eCos

سیستم عامل بلادرنگ eCos برای سیستم های جاسازی شده و کاربردهایی که تنها به یک فرایند با چند نخ نیاز دارند، استفاده می شود.

این سیستم عامل رایگان و کدهای آن قابل دسترسی است.

eCos به زبان برنامه نویسی C نوشته شده و لایه ها و واسط های برنامه سازی کاربردی آن مطابق با استاندارد های POSIX و μ ITRON می باشد.

این سیستم عامل دارای قابلیت پیکربندی است، بدین ترتیب می تواند مطابق نیاز های کاربردی مورد نظر، بهینه شود و با صدها گزینه اختیاری، بهترین عملکرد بلادرنگی و کمترین سخت افزار مورد نیاز ممکن را ارائه می کند.

این سیستم عامل برای استفاده در دستگاه هایی با فضای حافظه 10 تا 100 کیلوبایت طراحی شده است و بر روی طیف وسیعی از سخت افزاری شامل ARM، Motorola 68000، MIPS، PowerPC و غیره قابل اجرا می باشد.

برخی سیستم های مخابراتی ساخت ایران از این سیستم عامل استفاده می کنند.