

سیستم های کنترل بلادرنگ (جزوه دوم : تکمیلی جزوه اول)

سیستم های کنترل بلادرنگ: سیستم عامل بلادرنگ نوعی سیستم عامل است که در آن، زمان، پارامتر کلیدی است. برای مثال در سیستم‌های کنترل فرایند، رایانه های بی‌درنگ باید داده‌های فرآیند تولید را جمع‌آوری کرده و به کمک آن ماشینهای داخل کارخانه را کنترل کنند. سیستم‌هایی که آزمایش‌های علمی را کنترل می‌کنند، سیستم‌های تصویربرداری پزشکی، سیستم های کنترل صنعتی، کنترلرهای لوازم خانگی و سیستم های جنگ افزاری، سیستم‌های بی‌درنگ هستند.

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

مشخصات سیستم عاملهای بلادرنگ

قطعی بودن

پاسخدهی

کنترل کاربر

قابلیت اطمینان

نرمش با خطا

انواع سیستم های بلادرنگ

سیستم‌های Real-time به دو دسته تقسیم می‌شوند :

بلادرنگ سخت (Hard Real-time)

بلادرنگ نرم (Soft Real-time)

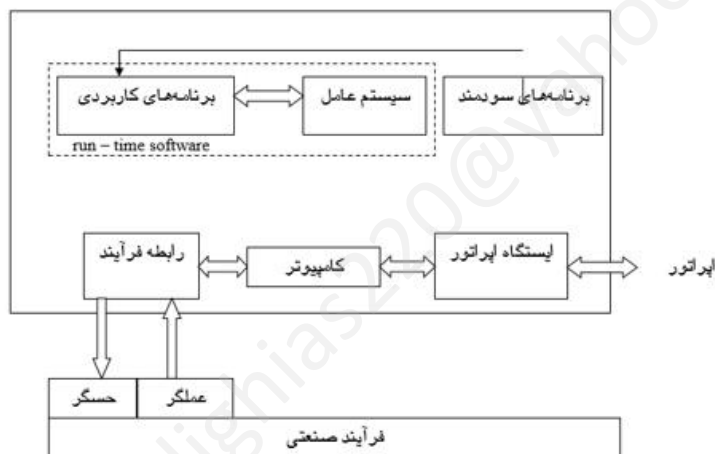
بلادرنگ سخت سیستمی است که در یک مهلت زمانی یا پاسخ میدهد یا هیچ. مانند سیستم مانیتورینگ بخش I.C.U یک بیمارستان، سیستم های کنترل صنعتی، موشکها، ماهواره ها...
سیستم بلادرنگ نرم سیستمی است که در بعضی از مواقع، آماده نشدن پاسخ در مهلت زمانی تعیین شده قابل تحمل است مانند مالتی مدیا

کنترل‌های بی‌درنگ

- 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

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

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 “دراپور” انجام می گیرد. دراپور 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 دراپورها 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 سیستم‌های بلادرنگ سه مرحله مورد بررسی است:

Scheduling زمانبندی
Programming برنامه‌نویسی
Verification بازبینی

زمانبندی

در یک سیستم بلادرنگ رویدادهایی رخ می‌دهد؛ برنامه به تعدادی پروسس تقسیم می‌شود 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 می‌شوند. رویداد غیر متناوب به صورت تصادفی رخ می‌دهند (زمان رخ داد مشخصی ندارند). از آنجا که پردازش مربوط به هر رویداد بخشی از زمان CPU را اشغال می‌کند، ممکن است پاسخ کلیه وقایع در مهلت مشخص امکان پذیر نباشد.

برنامه‌نویسی

تمامی سیستم‌های بلادرنگ دارای یک نرم‌افزار بلادرنگ هستند و این نرم‌افزار به وسیله یک زبان برنامه‌نویسی خاص تهیه می‌شود، در واقع یکی از راه‌های ایمن‌سازی سیستم‌ها، انتخاب زبان برنامه‌نویسی مناسب کارا است. عموماً از ویژگی‌های این زبان‌ها شامل سادگی، خوانایی، بدون ابهام بودن، ایمن بودن، مستقل از زمان بودن و قابل حمل بودن است.

همانطور که گفته شد این زبان‌ها بایستی بتوانند بلادرنگ عمل کنند و در نتیجه هر زبانی برای این کار مناسب نیست و در اینجا نمونه‌هایی از زبان‌های برنامه‌سازی بلادرنگ که از ابتدا تا کنون مورد استفاده قرار گرفته و می‌گیرد را مشاهده می‌کنید: Ada، Real-time Java، C، Erlang، Esterel، Real-time Euclid، Posix، Lustre، SAFL، Statecharts، Chill، Modula-2، Mesa، FRP، RTL/2، Coral 66، Jovial و ... است.

بازبینی

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

نتیجه گیری

در برخی کاربردها (مثل کنترل صنعتی) در کامپیوترها از سیستم عامل استفاده نمی‌شود. از آنجا که در سیستم‌های کنترل صنعتی برنامه می‌بایست در اسرع وقت در مقابل یک اتفاق، از خود عکس العمل نشان دهد، وجود واسطه سیستم عامل باعث کند شدن مراحل می‌گردد.

سیستم عامل بلادرنگ – RTOS

در اینجا پس از ارائه یک تعریف کلی از سیستم‌های بلادرنگ و کاربردهایشان، به بررسی الزام بکارگیری سیستم عامل بلادرنگ در چنین سیستم‌هایی می‌پردازیم. سپس ضمن بیان الزامات سیستم‌های عامل بلادرنگ و ویژگی‌های عمومی آنها، به بررسی پیش‌نیازهای 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 صحبت‌های خاتمه می‌دهم.

ارزیابی و مقایسه این سیستم‌های عامل، بستگی به مدیریت حافظه و زمانبندی در آنها دارد. سیستم بلادرنگ، نوعی سیستم کامپیوتری که صحت آن نه تنها به درستی منطق خروجی‌ها، 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. بطوری‌که اگر مهلت زمانی یک وظیفه رعایت نشود، نه تنها کیفیت عملکرد سیستم کاهش می‌یابد بلکه ممکن است منجر به لطمه‌های ناخواسته یا خطاهای غیر قابل برگشت به سیستم شود.

به عنوان مثال در یک نیروگاه هسته‌ای برخی کمیت‌ها باید همواره تحت کنترل دقیق باشند. مثلاً در یک راکتور هسته‌ای بر اثر پرتاب نوترون‌ها به اتم‌ها، نوترون‌های جدیدی آزاد می‌شوند و نوترون‌های آزاد شده نیز به اتم‌های دیگر برخورد کرده و نوترون‌های جدید آزاد می‌کنند و بهمین ترتیب. اگر تعداد نوترون‌های آزاد شده از یک حدی بیشتر شود، انفجار نوترونی رخ می‌دهد. پس غلظت نوترون‌ها 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 می‌توان سیستم مانیتورینگ بخش I.C.U بیمارستان را نام برد. یک تصور اشتباهی که درباره یک سیستم بلادرنگ وجود دارد این است که فکر می‌کنیم یک سیستم بلادرنگ، سیستمی با سرعت بالا است. در صورتیکه یک سیستم بلادرنگ الزماً یک سیستم با سرعت بالا نیست. یک سیستم می‌تواند خیلی کند کار کند، ولی در پاسخ دادن به رخدادها سرعت مناسبی داشته باشد.

سیستم‌های بلادرنگ

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

در گذشته، سیستم‌های کامپیوتری بلادرنگ تنها برای کاربرد های خاص طراحی می‌شدند، اما در حال حاضر سیستم‌های جاسازی شده قابل برنامه‌نویسی و پیکربندی، عمومیت یافته‌اند و در زمینه‌های مختلف بکار گرفته شده‌اند.

به عنوان نمونه ای از کاربردهای سیستم‌های بلادرنگ می‌توان از استفاده آنها در سیستم‌های کنترل نیروگاه‌ها، کنترل دستگاه‌های صنعتی، کنترل ترافیک هوایی، کنترل تجهیزات نظامی و موشک‌ها، کنترل نسبت مخلوط سوخت و هوا در کاربراتور خودرو، سیستم‌های تعویض خطوط راه آهن، سیستم‌های مخابراتی، اجاق مایکروویو و دستگاه‌های خودپرداز نام برد.

با توجه به کاربردهای سیستم‌های بلادرنگ و درجه حساسیت و فوریت واکنش به رخدادها خارجی، این سیستم‌ها به سه دسته-ی سیستم بلادرنگ سخت، سیستم بلادرنگ میان و سیستم بلادرنگ نرم تقسیم می‌شوند.

سیستم بلادرنگ سخت، سیستمی که برآورده نشدن محدودیت‌های زمانی در آن، باعث خراب شدن سیستم می‌شود. در چنین سیستمی ابداً نباید هیچ رخدادی از دست برود. مانند سیستم‌های کنترل شرایط بحرانی در نیروگاه‌های هسته‌ای

در این سیستم‌های بلادرنگ میان نیز باید قبل از پایان یافتن مهلت زمانی به رخدادهای سیستم، پاسخ داده شود. پاسخ بعد از مهلت زمانی، غیر ممکن یا بی‌فایده است. ولی پاسخ ندادن باعث خرابی‌های جدی نمی‌شود. مانند سیستم‌های پیش‌بینی وضع آب و هوا

سیستم بلادرنگ نرم، سیستمی که کیفیت عملکردش، بخاطر برآورده نشدن محدودیت‌های زمان پاسخگویی کاهش می‌یابد، اما باعث خراب شدن سیستم نمی‌شود. مانند سیستم‌های مخابراتی

در سیستم‌های بلادرنگ بسته به سطح پیچیدگی عملکرد سیستم ممکن است از یک سیستم عامل بهره بگیریم. سیستم عامل غالباً در مواردی استفاده می‌شود که پیچیدگی عملکرد سیستم، یا نیاز آن به پروتکل‌های ارتباطی، وجود آن را اجباری کند، وگرنه در سیستم‌های 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:

مشخصات سیستم عاملهای بلادرنگ

قطعی بودن پاسخدهی کنترل کاربر قابلیت اطمینان نرمش با خطا

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 نخ‌های متعدد بتوانند بطور منظم و در زمان مناسب با یکدیگر ارتباط برقرار کنند، مکانیزم‌هایی برای ارتباط بین وظایف و همگام سازی با قابلیت پیش بینی مورد نیاز هستند. قابلیت پیش بینی پذیری نیز توسط حفظ یکپارچگی و صحت داده‌ها و رعایت ترتیب اجرا به موقع وظایف امکان پذیر است. یکی از روش‌های حفظ یکپارچگی داده‌ها، قابلیت قفل کردن منابع می‌باشد.

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 می‌دهد 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، می‌پردازیم.

Windows CE

نسخه ای از سیستم عامل Windows شرکت Microsoft، برای کامپیوترهای کوچک و سیستم‌های جاسازی شده 32 بیتی سیار می‌باشد.

واضح است که Windows CE دارای هسته ای متفاوت از Windows‌های رومیزی می‌باشد.

این سیستم عامل روی پردازنده‌های Intel x86 پشتیبانی می‌شود و سازگار با پردازنده‌های ARM، MIPS و Hitachi SuperH می‌باشد.

این سیستم عامل دارای 256 سطح الویت است.

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

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 LynuxWorks ارائه می شود.

این سیستم عامل منطبق با استاندارد POSIX و سازگار با Linux باشد و دارای ویژگی چند نخه 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 فضانوردی، 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 مبنی 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.

مزایای اتوماسیون صنعتی:

- 1) تکرارپذیری فعالیتها و فرایندها
- 2) افزایش کیفیت محصولات تولیدی
- 3) افزایش سرعت تولید (افزایش کمیت تولید)
- 4) کنترل کیفیت دقیقتر و سریعتر
- 5) کاهش پسماندهای تولید (ضایعات)
- 6) واکنش های متقابل بهتر با سیستمهای بازرگانی و اداری و مالی و منابع انسانی
- 7) افزایش بهره وری واحدهای صنعتی
- 8) بالا بردن ضریب ایمنی برای نیروی انسانی و کاستن از فشارهای روحی و جسمی

امتیازات PLC در مقایسه با کنترل کننده های رله ای و کنتاکتوری

- 1) هزینه نصب و راه اندازی آنها پایین می باشد.
- 2) برای نصب و راه اندازی آنها زمان کمتری لازم است .
- 3) اندازه فیزیکی کمی دارند.
- 4) تعمیر و نگه داری آنها بسیار ساده می باشد.
- 5) به سادگی قابلیت گسترش دارند .
- 6) قابلیت انجام عملیات پیچیده را دارند.
- 7) ضریب اطمینان بالایی در اجرای فرایندهای کنترلی دارند .
- 8) ساختار مدولار دارند که تعویض بخشهای مختلف آن را ساده میکند.
- 9) اتصالات ورودی - خروجی و سطوح سیگنال استاندارد دارند.
- 10) زبان برنامه نویسی آنها ساده و سطح بالاست.
- 11) در مقابل نویز و اختلالات محیطی حفاظت شده اند.
- 12) تغییر برنامه در هنگام کار آسان است.
- 13) امکان ایجاد شبکه بین چندین PLC به سادگی میسر است .
- 14) امکان کنترل از راه دور (به عنوان مثال از طریق خط تلفن یا سایر شبکه های ارتباطی) قابل حصول است .
- 15) امکان اتصال بسیاری از تجهیزات جانبی استاندارد از قبیل چاپگر ، بارکد خوان و ... به PLC ها وجود دارد .

اهداف مونیتورینگ

- 1) عبارت است از جمع آوری اطلاعات مورد نظر از بخشهای مختلف یک واحد صنعتی و نمایش آنها با فرمت مورد نظر برای رسیدن به :
- 2) نمایش وضعیت لحظه ای هر یک از ماشین آلات و دستگاهها
- 3) نمایش و ثبت پارمترهای مهم و حیاتی یک سیستم
- 4) نمایش و ثبت آلامهای مختلف در زمانهای بروز خطا در سیستم
- 5) نمایش محل خرابی و زمان وقوع ایراد در هر یک از اجزای سیستم
- 6) نمایش پروسه های تولید با استفاده از ابزارهای گرافیکی مناسب
- 7) تغییر و اصلاح Set Point ها حین اجرای پروسه تولید
- 8) امکان تغییر برخی از فرایندهای کنترلی از طریق برنامه مونیتورینگ
- 9) ثبت اطلاعات و پارامترهای مورد نظر مدیران از قبیل زمانهای کارکرد، میزان تولید ، میزان مواد اولیه مصرفی ، میزان انرژی مصرفی و ..

نمونه ای از دستاوردهای حاصل از اتوماسیون برق

- 1) نمایش وضعیت لحظه ای هر یک کلیدها و بریکرها و تپ چنجر و ..
- 2) نمایش و ثبت مقادیر جریان و ولتاژها و بار اکتیو و راکتیو و در ورودی و خروجیها و دیگر محل ها و ..
- 3) نمایش و ثبت آلامهای مختلف در زمانهای بروز خطا و ..
- 4) الحاق برچسب زمانی Time Tag به وضعیت ها و آلامها و اندازه گیریها و فرامین
- 5) استفاده از برچسب زمانی برای مشخص نمودن اولین عملکرد و عملکردهای بعدی، در زمانی که یک حادثه منجر به گسترش حادثه شده است
- 6) استفاده از رنگها و پیام صوتی و نوشتاری برای اعلام هشدار
- 7) ثبت اطلاعات و گزارشگیری متنوع جهت تصمیم گیری مدیریتی
- 8) سیستم مدیریت انرژی (EMS = Energy Management System)
- 9) امکان پیش بینی اثرات تغییرات آینده قبل از اجرای واقعی آن (PAS=Power Advanced Software)
- 10) امکان بررسی وضعیت حوادث رخ داده در گذشته، با برگرداندن سیستم فعلی بصورت مجازی به شرایط قبل از حادثه (پست مورتم)
- 11) شبیه ساز آموزشی
- 12) مطالعه بار (جریان) Load Flow Study

DTS=Dispatching Training Simulator

OTS=Operator Training Simulator

مطالعه بار (جریان) Load Flow Study