سیستم های بلادرنگ

استاد: امین صدیقی
شیراز 1394

Sedighias220@yahoo.com

اتوماسیون

سیستم های بلادرنگ

سیستم عامل های بلادرنگ

انواع سیستمها

معیارهای انتخاب

انواع زمانبندی سیستم ها

پیوست:

مدل OSI

سنسور

ترانسدیوسر

ترانسمیتر

انکودرها

روباتها

میکروکنترلر

مقدمه:

اتوماسیون چیست

(Automation)ترکیبی است از دو کلمه "Automatic" و "Operation" و به معنی عمل کردن بدون عامل خارجی (انسان) یا با کمترین تاثیر عامل خارجی است. بعبارتی اتوماسیون یعنی خودکارشدن عملیات.

خودکارسازی اداری مجموعه‌ای از روش‌ها برای ثبت و نمایش و مدیریت روندهای اداری , فنی و مالی میباشد. اغلب این راهکارها بر پایه این جمله بوجود آمده اند

سرعت پایین، افزایش بوروکراسی غیر نیاز ، وابسته شدن سیستم به افراد و نیز عدم هماهنگی روش سنتی با نرم افزارها سبب افزایش قابل ملاحظه خطاهای انسانی و کاهش بهره وری در سازمانها گردید و در نتیجه

سازمانها تمایل به اتوماسیون اداری و مالی و فنی، یکپارچه، گرفتند که منجر به پیدایش Management Information System و باختصار MIS گردید

اتوماسیون اداری Office Automation

اتوماسیون اداری دربرگیرنده سیستم های مالی , پرسنلی , اداری , بایگانی و فنی میباشد و در یک مجموعه متمرکز و مرتبط قابل استفاده میباشد (سیستم ERP - Enterprise Resource Planning).

برنامه ریزی، سازماندهی، کنترل و نظارت بر عملکرد فعالیتهای درون سازمانی یکی از مهمترین معیارها و پیش نیازها در توسعه و کاربرد فناوری اطلاعات در سازمانهای امروزی محسوب میشود.

سازمانهای سنتی حتی موسسات موفق و مطرح جامعه , عملا تنها از 30درصد منابع انسانی خود استفاده میکنند و از 70درصد دیگر غیر از اتلاف وقت و سربار هزینه چیزی عایدشان نمی‌شود.

در اتوماسیون اداری و مالی از سخت افزار و نرم افزارها الکترونیکی و دیجیتالی متنوعی، برای ایجاد، جمع آوری، ذخیره و تغییر داده های اداری به اطلاعات اداری مورد نیاز برای انجام کارهای اصلی سازمان، استفاده میشود.

اتوماسیون اداری و مالی نه تنها به خودکار سازی و بهینه سازی رویه‌های جاری سازمان‌ها با دقت و سرعت بسیار زیاد کمک می‌کند، بلکه میتواند با توجه به آرشیو و سوابق اطلاعات بشکل هوشمند تصمیم سازی برای آینده را ارائه دهد.

اتوماسیون صنعتی

اتوماسیون صنعتی به بهره گیری از نرم‌افزار و سخت‌افزار ها، بجای متصدیان انسانی برای کنترل دستگاه ها و فرایندهای صنعتی گفته میشود.

اتوماسیون یک گام فراتر از مکانیزه کردن است . مکانیزه کردن به معنی فراهم کردن متصدیان انسانی با ابزار و دستگاه هایی است که ایشان را برای انجام بهتر کارشان یاری میرساند .نمایانترین و شناخته شده ترین بخش اتوماسیون صنعتی ربات های صنعتی هستند .

امروزه کاربرد اتوماسیون صنعتی و ابزار دقیق در صنایع و پروسه های مختلف صنعتی به وفور به چشم میخورد . کنترل پروسه و سیستمهای اندازه گیری پیچیده ای که در صنایعی همچون آب، برق، نفت ، گاز ، پتروشیمی ، صنایع شیمیایی ، صنایع غذایی ، صنایع خودرو سازی و غیره بکار می آید نیازمند ابزارآلات بسیار دقیق و حساس می باشند . پیشرفتهای تکنیکی اخیر در کنترل فرایند و اندازه گیری پارامترهای مختلف صنعتی از قبیل فشار ، دما ، فلوی آب، جریان، ولتاژ، وضعیت کلیدها و دریچه ها و غیره باعث افزایش کیفیت محصولات و کاهش هزینه های تولید گردیده است .

بعضی از 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

لازمه افزایش کیفیت و کمیت یک محصول ، استفاده از ماشین آلات پیشرفته و اتوماتیک می باشد . ماشین‌آلاتی که بیشتر مراحل کاری آنها به طور خودکار صورت گرفته و اتکای آن به عوامل انسانی کمتر باشد.

چنین ماشین آلاتی جهت کارکرد صحیح خود نیاز به یک بخش فرمان خودکار دارند که معمولا از یک سیستم کنترل قابل برنامه ریزی (به عنوان مثال PLC=Programable Logic Control یامدار منطقی قابل برنامه ریزی) استفاده میکنند.

جمع آوری اطلاعات در فرایندهای صنعتی با استفاده از سنسورها یا حسگرها صورت می گیرد . این حسگرها به منزله چشم و گوش یک سیستم کنترلی عمل می کنند . امروزه در بسیاری از ماشین آلات صنعتی استفاده از سنسورها امری متداول می باشد تا جاییکه عملکرد خودکار یک ماشین را می توان با

تعداد سنسورهای موجود در آن درجه بندی کرد . وجود سنسورهای مختلف در فرایند اتوماسیون به اندازه ای مهم می باشد که بدون سنسور هیچ فرایند خودکاری شکل نمی گیرد بنابراین سنسورها یکی از اجزای لاینفک سیستمهای اتوماسیون صنعتی می باشند .

در گذشته نه چندان دور بسیاری از تابلوهای فرمان ماشین آلات صنعتی ، برای کنترل پروسه های تولید از رله های الکترومکانیکی یا سیستمهای پنوماتیکی استفاده می کردند و اغلب با ترکیب رله های متعدد و اتصال آنها به یکدیگر منطق کنترل ایجاد می گردید . در بیشتر ماشین آلات صنعتی ، سیستمهای تاخیری و شمارنده ها نیز استفاده میگردید و با اضافه شدن تعدادی Timer و شمارنده به تابلوهای کنترل حجم و زمان مونتاژ آن افزایش می یافت .

اشکال فوق با در نظر گرفتن استهلاک و هزینه بالای خود و همچنین عدم امکان تغییر در عملکرد سیستم ، باعث گردید تا از دهه 80 میلادی به بعد اکثر تابلوهای فرمان با سیستمهای کنترلی قابل برنامه ریزی جدید یعنی PLC جایگزین گردند .در حال حاضر PLC یکی از اجزای اصلی و مهم در پروژه های اتوماسیون می باشد که توسط کمپانیهای متعدد و در تنوع زیاد تولید و عرضه میگردد . به طور خلاصه سیستمهای نوین اتوماسیون و ابزار دقیق مبتنی بر PLC در مقایسه با کنترل کننده های رله ای و کنتاکتوری قدیمی دارای امتیازات زیر است :

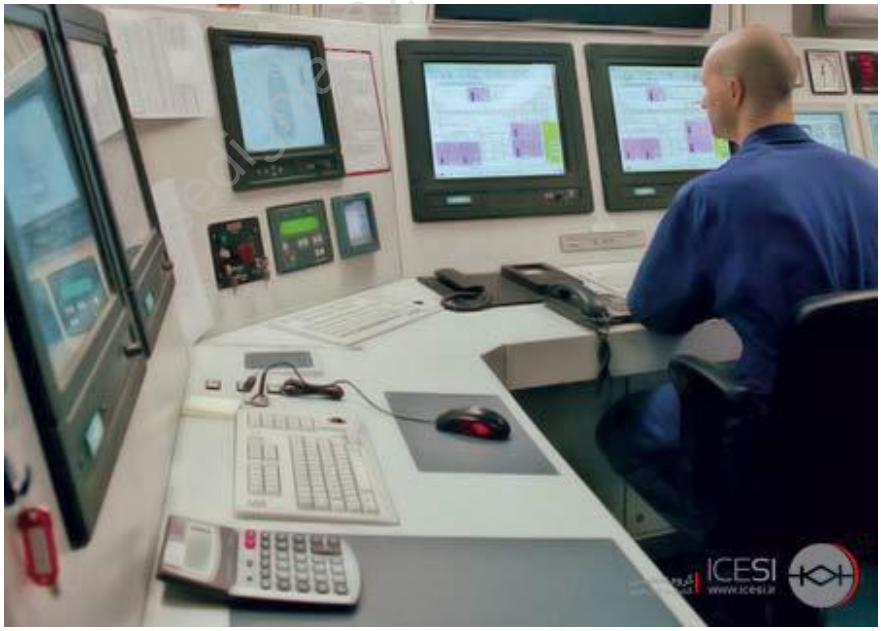
1) هزینه نصب و راه اندازی آنها پایین می باشد.
2) برای نصب و راه اندازی آنها زمان کمتری لازم است .
3) اندازه فیزیکی کمی دارند.
4) تعمیر و نگه داری آنها بسیار ساده می باشد.
5) به سادگی قابلیت گسترش دارند .
6) قابلیت انجام عملیات پیچیده را دارند.
7) ضریب اطمینان بالایی در اجرای فرایندهای کنترلی دارند .
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 .

مونیتورینگ در اتوماسیون

یکی دیگر از مباحث مهم و مرتبط با اتوماسیون صنعتی ، مانیتورینگ می باشد . امروزه مانیتورینگ یکی از نیازهای اساسی بسیاری از صنایع به خصوص صنایع بزرگ می باشد. بسیاری از صنایع بزرگ مانند صنایع برق، آب، پتروشیمی ، صنایع تولید انرژی ، صنایع شیمیایی و ... بدون استفاده از سیستم مونیتورینگ مناسب قادر به ادامه کار خود نیستند .

مونیتورینگ عبارت است از جمع آوری اطلاعات مورد نظر از بخشهای مختلف یک واحد صنعتی و نمایش آنها با فرمت مورد نظر برای رسیدن به اهداف ذیل :

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



مفاهیم در کنترل صنعتی

سنسور دمای RTD

مقاومت الکتریکی یک جسم با دمای آن رابطه دارد. مقاومت الکتریکی بسیاری از فلزات، با افزایش دما افزایش و با کاهش آن کاهش میابد. عملکرد RTD ها بر اساس همین موضوع میباشد.
به عبارت دیگر در یک RTD با اندازه گیری مقاومت یک فلز، دمای آن تعیین میشود.

لودسل

سنسوری الکترونیکی هست که جهت اندازه گیری نیرو و وزن اجسام به کار گرفته میشود

ترانسمیتر:

ترانسمیتر از ترکیب دوکلمه ای انتقال (Transfer) و اندازه گیری (Metering) تشکیل شده است و به معنی تجهیزی می باشد که بتواند کمیت فیزیکی را اندازه گیری کرده و سپس سیگنال اندازه‌گیری شده را برای کنترل کننده ارسال نماید.
ترانسدیوسر – یک نمونه از ترانسمیتر میباشد
ترانسمیترها الکترونیکی و یا نیوماتیکی می باشندکه در هر دوحالت، سیگنالی استاندارد را ارسال می نمایند که برای تجهیزاتی که در LOOP کنترل قراردارند، قابل فهم می باشد. در ترانسمیتر های نوع الکترونیکی جریان ۴ تا ۲۰ میلی آمپر و درنوع نیوماتیکی فشار هوای ۳ تا ۱۵ PSI از سوی ترانسمیتر به کنترلرهای الکترونیکی و یا نیوماتیکی ارسال می شود.

ترانسمیتر های جریان (ترانسدیوسر جریانی)

با نصب ترانس های CT در داخل تابلو های قدرت، نمونه جریان اندازه گیری میشود که میتوان مقدار آنرا روی یک نمایشگر مشاهده نمود
ترانسمیتر های جریان CT نمونه ای از یک ترانس CT می باشد که ورودی آن جریانی مثلا صفر تا یک آمپر یا صفر تا 5 آمپر بوده و خروجی آن سیگنال4تا20میلی آمپر میباشد .
از ترانسمیتر برای اندازه گیری و و ذخیره و مونیتورینگ. جریان ورودی و خروجی یک سیستم (تابلو یا موتورها) استفاده میشود که خروجی ان براحتی به کنترلر های مختلف نظیر PLC و نمایشگر ها وصل میشود

پوزیشنر

پوزیشـنر یک دستگاه کنترل حرکت است که به صورت مداوم موقعیت یک تجهیز(مثلا یک دریچه) را با سـیگنال ارسـالی مقایسه میکند و تا زمانی موقعیت تجهیز با موقعیت درخواستی (که توسط سیگنال الکتریکی ارسـال شـده) برابر نشـده باشـد فشـار ارسالی برای دیافراگم و یا پیستون اکچویتور را تنظیم میکند تا موقعیت دریچه با موقعیت ارسالی یکی شود.

عملگر Actuator

همان عنصـری هست که در آخر هر سیستم کنترلی قرار گرفته و عملیات و فعالیت مورد نظر را انجام می دهد

اینورتر VFD

اینورتر (VFD) یا VARIABLE FREQUENCY DRIVE نوعی کنترل کننده و راه انداز موتور است. که با تغییر دادن فرکانس و ولتاژ اعمال شده به الکتروموتور آن را به گردش در می آورد یا آن را راه اندازی میکند.

سرو موتور:

یک نوع موتور الکتریکی میباشد که به دلیل استفاده در پروژه های صنعتی به صورت حلقه بسته ، مجهز به سیستم های کنترل فیدبک دار شده است ، که متغییر کنترل شونده موقعیت، سرعت،گشتاور میباشد.لختی در این موتور ها بسیار پایین بوده و درنتیجه تغییر سرعت در این نوع موتور های بسیار سریع میباشد.

سروو موتور بصورت AC و DC میباشد،

سروو موتور هایAC به دو صورت ۳ فاز و تک فاز میباشد

از سـرو موتور در تمام پروسه های کنترلی و عمومی که نیاز به موتور الکتریکی میباشد استفاده میشود (مثلا در پروسه -کنترل موقعیت -کنترل سرعت -کنترل گشتاور -کنترل دستی)

مانیتورینگ صنعتی

امروزه در کارخانجات و واحد های بزرگ صنعتی راهبری سیستمها و ماشین آلات به صورت محلی و توسط اپراتور خاص برای هر قسمت،به علت گستردگی سایت فاصله بین یونیت ها،حجم بالای تجهیزات و عدم امکان ایجاد هماهنگی های مورد نیاز بین واحد های مختلف،امکان پذیر نیست.

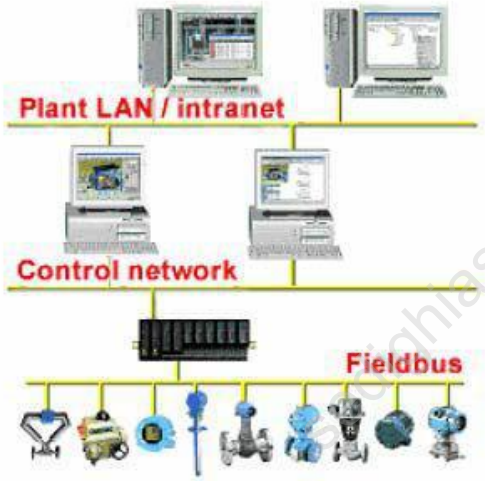
از مجموعه ای به نام سیستم کنترل و مانیتورینگ استفاده میشود.این مجموعه شامل دو بخش زیر میباشد :

سیستم کنترل:

متشکل از PLC/DCS و سخت افزارهای مربوطه که مستقیم با I/O های داخل فیلد در ارتباط هستند.

سیستم Monitoring و Operating :

این مجموعه در اتاقی به نام اتاق مرکزی یا CCR که مخفف Central control Room میباشد، نصب میگردد و اپراتورها از طریق این واسط ها کنترل واحد ها را انجام میدهند.از آنجاییکه این سیستم ها رابط بین کاربر و ماشین آلات موجود در سایت میباشند.به آنها اجمالا HMI یا رابط بین ماشین و انسان گفته میشود.



HMI : رابط بین ماشین و انسان است، سیستمی است که کاربر(اپراتور) میتواند یک صنعت را مشاهده و یا با فرمان کنترل نماید

نکته:

دو نام مشابه ولی با مفهوم متفاوت

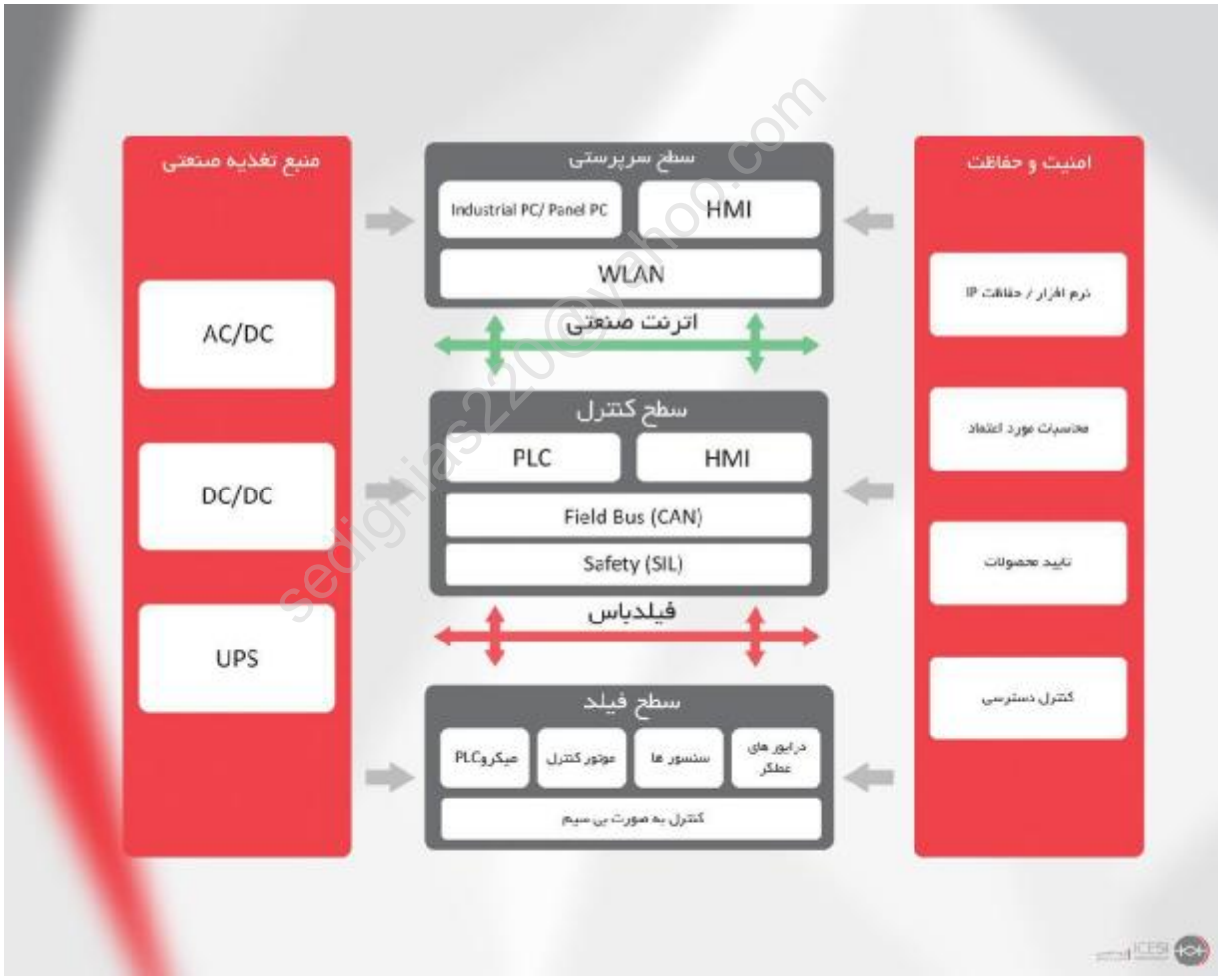
PLC = Programmable Logic Control مدارات قابل برنامه ریزی دیجیتالی

PLC = Power Line Carrier ارسال سیگنال (صوت و داده) روی سیم شبکه برق

آیا در یک سیستم صنعتی انتخاب مشخصات فنی سنسورها چگونه باید باشد و حسگرها(سنسورها) – و مسیر مخابرات داده ها و انتخاب سخت‌افزارها و کامپیوترها و سرورها چه اهمیتی دارد سرعت سخت‌افزار و انتخاب سیستم عامل و نرم‌افزار و سرعت سیستم عامل و سرعت نرم‌افزار و امنیت اطلاعات و صفر شدن خطا چه اهمیتی دارد

اگر در یک سیستم صنعتی مثلا همزمان چندین سنسور دما و سنسور فشار همزمان اطلاعاتی را حس کنند از چه مسیر(مدیای) مخابراتی و به چه سخت افزار و نرم افزار ارسال کنند و سپس سیستم اتوماسیون با چه تحلیلی و با چه سرعتی و با چه امنیتی فرامین صحیح به عملگرها (اکچویتورها) ارسال کنند.

یکی از مسائل مهم در صنعت عکس العمل بلادرنگ سیستمها میباشد



RealTime Systems

A. Definitions
B. Role of an OS in Real Time Systems
C. Features of Real Time Operating Systems (RTOS)
   - Scheduling
   - Resource Allocation
   - Interrupt Handling
   - Other Issues
D. Linux for Real Time Systems and RTLinux
E. Other RTOS's

A. Definitions

REAL TIME SYSTEM (RTS)

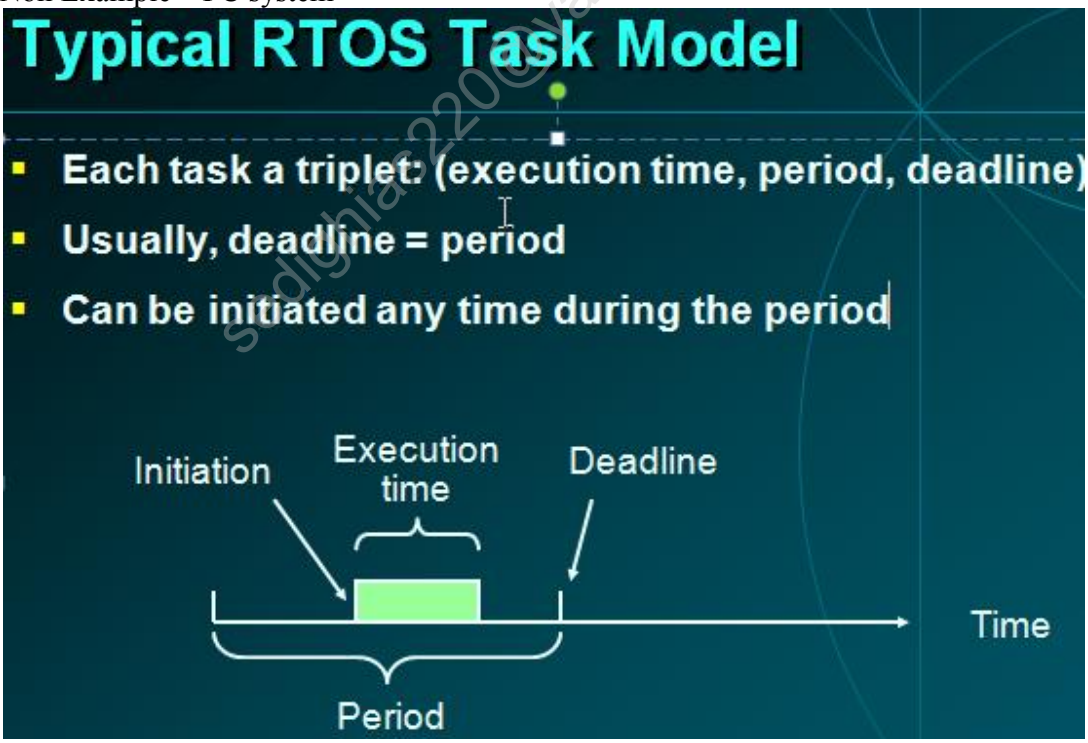
A real time system is one in which the correctness of the computations not only depends on their logical correctness, but also on the time at which the result is produced

A system is said to be Real Time if it is required to complete it's work & deliver it's services on time.

Example – Flight Control System

All tasks in that system must execute on time.

Non Example – PC system



A Hard RTS is one in which one or more activities must never miss a deadline or timing constraints, otherwise the system fails or results in catastrophe. [St]

- Hard Real Time System
  - Failure to meet deadlines is fatal
    - example : Flight Control System

- Hard Real Time System
  - Validation by provably correct procedures or extensive simulation that the system always meets the timings constraints

A Soft RTS is one that has timing constraints, but occasionally missing them has negligible effects, as application requirements as a whole continue to be met

- Soft Real Time System
  - Late completion of jobs is undesirable but not fatal.
  - System performance degrades as more & more jobs miss deadlines
  - Online Databases
- Soft Real Time System
  - Demonstration of jobs meeting some statistical constraints suffices.
- Example – Multimedia System
  - 25 frames per second on an average

B. Role of an OS (Operating Systems) in Real Time Systems

1. Standalone Applications
   - Often no OS involved
   - Micro controller based Embedded Systems
2. Some Real Time Applications are huge & complex
   - Multiple threads
   - Complicated Synchronization Requirements
   - Filesystem / Network / Windowing support
   - OS primitives reduce the software design time

C. Features of RTOS (Real Time Operating Systems)

1. Scheduling

1.1. More information about the tasks are known

- No of tasks
- Resource Requirements
- Release Time
- Execution time
- Deadlines

1.2. Being a more deterministic system better scheduling algorithms can be devised.

Scheduling Algorithms in RTOS

- Clock Driven Scheduling
  - All parameters about jobs (release time/ execution time/deadline) known in advance.
  - Schedule can be computed offline or at some regular time instances.
  - Minimal runtime overhead.
  - Not suitable for many applications.
- Weighted Round Robin Scheduling
  - Jobs scheduled in FIFO manner

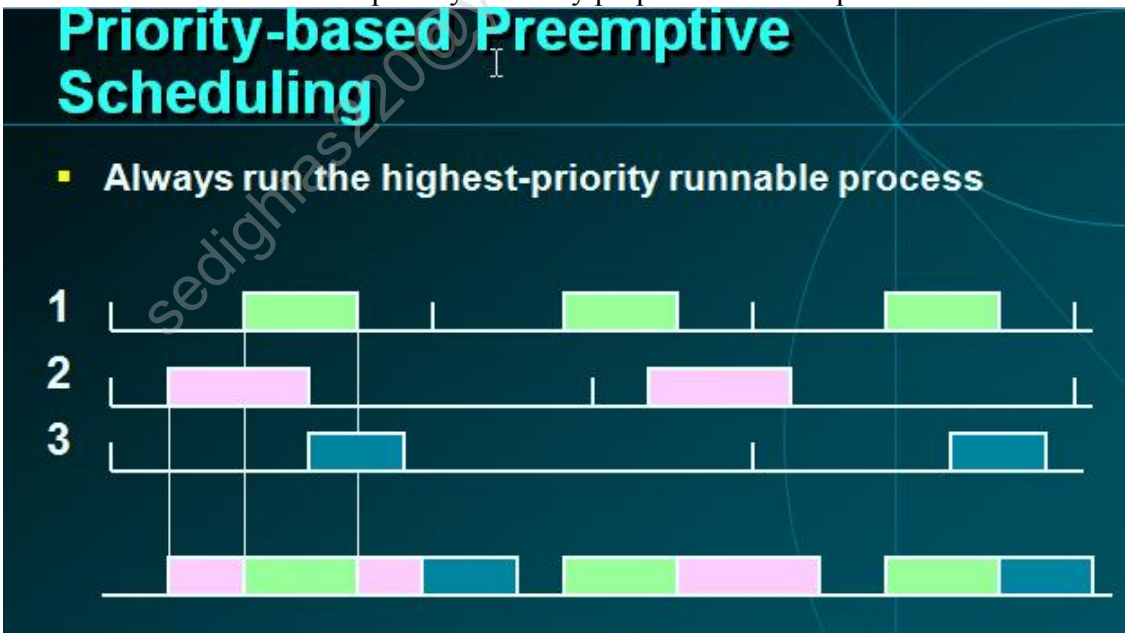
- Time quantum given to jobs is proportional to it's weight
- Example use : High speed switching network
- QOS guarantee.
- Not suitable for precedence constrained jobs.
  - Job A can run only after Job B. No point in giving time quantum to Job B before Job A.

- Priority Scheduling
  (Greedy/List/Event Driven)
  - Processor never left idle when there are ready tasks
  - Processor allocated to processes according to priorities
  - Priorities
    - static - at design time
    - Dynamic - at runtime

Priority Scheduling
Earliest Deadline First (EDF)
- Process with earliest deadline given highest priority

Least Slack Time First (LSF)
- slack = relative deadline – execution left

Rate Monotonic Scheduling (RMS)
- For periodic tasks
- Tasks priority inversely proportional to it's period



2. Resource Allocation in RTOS
   - Resource Allocation
     - The issues with scheduling applicable here.
     - Resources can be allocated in
       - Weighted Round Robin
       - Priority Based
   - Some resources are non preemptible

- Example : semaphores
- Priority Inversion if priority scheduling is used

Solutions to Priority Inversion

- Non Blocking Critical Section
  - Higher priority Thread may get blocked by unrelated low priority thread
- Priority Ceiling
  - Each resource has an assigned priority
  - Priority of thread is the highest of all priorities of the resources it's holding
- Priority Inheritance
  - The thread holding a resource inherits the priority of the thread blocked on that resource

3. Interrupt Handling in Linux
   - Interrupts are disabled in ISR/critical sections of the kernel
   - No worst case bound on interrupt latency avaliable
     - eg: Disk Drivers may disable interrupt for few hundred milliseconds
   - Not suitable for Real Time Applications
     - Interrupts may be missed

D. Linux for Real Time Systems and RTLinux

- Scheduling
  - Priority Driven Approach
    - Optimize average case response time
  - Interactive Processes Given Highest Priority
    - Aim to reduce response times of processes
  - Real Time Processes
    - Processes with high priority
    - No notion of deadlines
- Resource Allocation
  - No support for handling priority inversion

Other Problems with Linux Beside Interrupt Handling in Linux

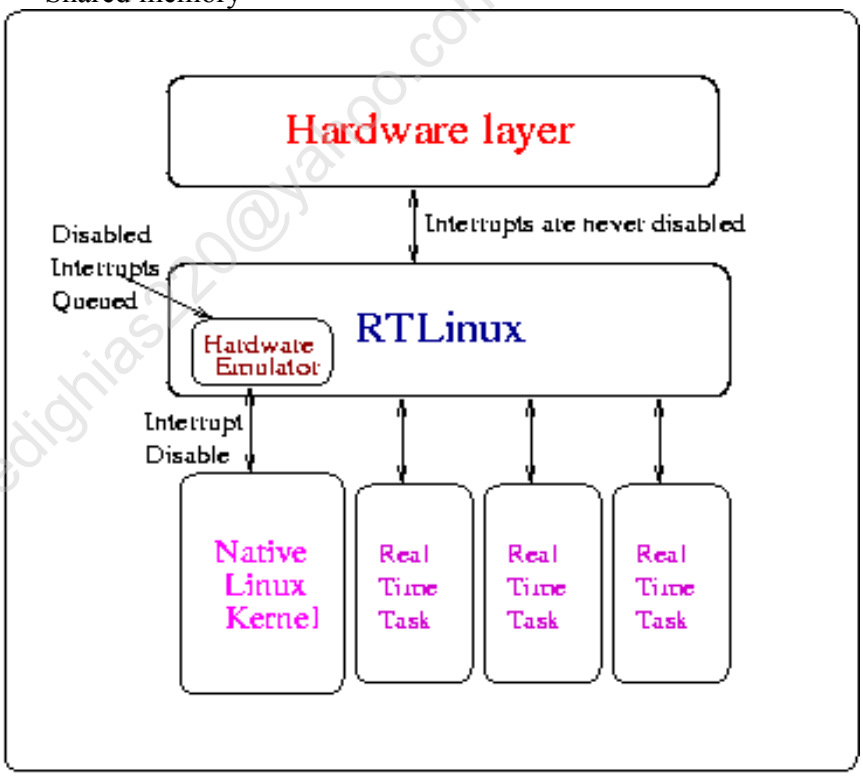
- Processes are non preemtible in Kernel Mode
  - System calls like fork take a lot of time
  - High priority thread might wait for a low priority thread to complete it's system call
- Processes are heavy weight
  - Context switch takes several hundred microseconds

Why Linux

- Coexistence of Real Time Applications with non Real Time Ones
  - Example http server
- Device Driver Base
- Stability

RTLinux

- Real Time Kernel at the lowest level
- Linux Kernel is a low priority thread
  - Executed only when no real time tasks
- Interrupts trapped by the Real Time Kernel and passed onto Linux Kernel
  - Software emulation to hardware interrupts
    - Interrupts are queued by RTLinux
    - Software emulation to disable_interrupt()
- Real Time Tasks
  - Statically allocate memory
  - No address space protection
- Non Real Time Tasks are developed in Linux
- Communication
  - Queues
  - Shared memory



E. Other RTOS Issues

- Interrupt Latency should be very small
  - Kernel has to respond to real time events
  - Interrupts should be disabled for minimum possible time
- For embedded applications Kernel Size should be small
  - Should fit in ROM
- Sophisticated features can be removed

- No Virtual Memory
- No Protection

- Some Good features in choosing RTOS
- Motivation
  - Full grasp over source code – Easily modifiable, portable
- Features
  - Modular Design
    - Isolation of Architecture/CPU dependent and independent code – Easy to Port
  - Pluggable Scheduler
    - Scheduler - part of the Application
    - Kernel interacts with the scheduler through an API
    - Application developer needs to implement the scheduler API
  - Can optimize on Data Structures & Algorithms for implementing the scheduler
  - More than one level Interrupt Handling
    - Two level Interrupt Handling
      - Top Half Interrupt Handler
        - Called Immediately – Kernel never disables interrupts
        - Cannot invoke thread library functions - Race Conditions
      - Bottom Half Interrupt Handler
        - Invoked when kernel not in Critical Section
        - Can invoke thread library functions
    - Very Low Response time
  - Small footprint
    - Small footprint (~50kb)
  - Oskit's Device Driver Framework
    - Allows direct porting of existing drivers from Linux.
    - Example – Ethernet Driver of Linux

Other RTOS's

- LynxOS
  - Microkernel Architecture
    - Kernel provides scheduling/interrupt handling
  - Additional features through Kernel Plug Ins(KPIs)
    - TCP/IP stack , Filesystem
    - KPI's are multithreaded
  - Memory Protection/ Demand Paging Optional
  - Development and Deployment on the same host
    - OS support for compilers/debuggers
- VxWorks
  - Monolithic Architecture
  - Real Time Posix compliant
  - Cross development System

- pSOS
  - Object Oriented OS

F. Peripheral devices and protocols

- Interfacing
  - Serial/parallel ports, USB, I2C, PCMCIA, IDE
- Communication
  - Serial, Ethernet, Low bandwidth radio, IrDA,
  - 802.11b based devices
- User Interface
  - LCD, Keyboard, Touch sensors, Sound, Digital
  - pads, Webcams
- Sensors
  - A variety of sensors using fire, temperature,
  - pressure, water level, seismic, sound, vision

مقدمه

آیا می‌دانید رابطه‌ی کامپیوتر با سیستم‌عامل چیست؟

طراحی سیستم های بلادرنگ همانند صنعت نرم افزار یک موضوع رو به افزایش در انجمنهای پژوهش های سیستم ها است درخواست های بلادرنگ و نیازهای آن ها را می‌توان تقریباً در محدوده ی تمام تحقیقات و پژوهش های سیستم عامل و شبکه یافت

در علم کامپیوتر، محاسبات بلادرنگ موضوعی از سیستم های سخت افزار و نرم افزار است که در رابطه با قید زمانی است.

کاربردهای استاندارد سیستم بلادرنگ

سیستم کنترل دیجیتال

سیستم فرمان و کنترل

پردازش سیگنال

سیستم ارتباطات راه دور

کاربردهای جدید سیستمهای بلادرنگ

شبکه بلادرنگ

سیستم های بلادرنگ سخت و نرم

سیستم عاملهای بلادرنگ

علت رویکرد به سیستم بلادرنگ

افزایش تعداد کاربردهای توزیعی بلادرنگ در فراهم کردن سرویس های ارتباطی قابل پیش بینی، منفعت زیادی

مثالهایی از کاربردهای شبکه بلادرنگ

هنگام رانندگی

در هنگام پرواز

در هنگام سرماخوردگی

سیستم های بلادرنگ سخت و نرم

سیستم بلادرنگ سخت

سیستم های بلادرنگ نرم

مدل های وظیفه در سیستمهای بلادرنگ

وظیفه های بلادرنگ دوره ای

وظیفه های بلادرنگ نامنظم

وظیفه های بلادرنگ دوره ای

مثال یک درخواست روباتیک

وظیفه های بلادرنگ نامنظم

نمونه های وظیفه های غیر تناوبی در سیستم های بلادرنگ با رویداد هدایت شده مانند بیرون پرتاب شدن صندلی خلبان هنگامی که فرمان به سیستم ناوبری جت داده می شود است.

سیستمهای بلادرنگ

"به سیستمی بلادرنگ گفته می شود که صحت درستی یک فرایند تنها وابسته به صحت منطقی نباشد، بلکه به زمانی که در آن اجرا می شود نیز وابسته باشد."

در علم کامپیوتر، محاسبات بلادرنگ موضوعی از سیستم های سخت افزار و نرم افزار است که در رابطه با قید زمانی است. مثل پاسخگویی به حوادثی که ناشی از حساسیت های زمانی هستند. در مقابل سیستم های غیر بلادرنگ هستند که محدودیت زمانی ندارند، حتی اگر سرعت پاسخگویی و اجرا مطلوب یا رویدادی ارجح باشد.

کامپیوترها و شبکه هایی که از سیستم بلادرنگ استفاده می کنند، بر خلاف کامپیوترهای شخصی و سیستم های کامپیوتری، که مجری برنامه های غیر-بلادرنگ ازقبیل مرورگر شبکه هستند، از دید کاربر مخفی هستند.و طوری به نظر میرسد که این سیستم ها وجود ندارند.

نیاز نرم افزارهای بلادرنگ معمولاً آدرس دهی در فضای سیستم است و زبان های برنامه نویسی، هم زمانی را که چارچوب نرم افزارهای بلادرنگ را می سازد، فراهم می کند.

سیستم ضد قفل در ترمز ماشین نمونه ساده‌ای از سیستم های بلادرنگ است. محدودیت زمانی در این سیستم، زمان کوتاهی است که ترمز باید گرفته شود، تا از قفل شدن چرخ ها جلوگیری شود.

محاسبات بلادرنگ اگر قبل از محدودیت زمانی، جایی که این محدودیت مربوط به یک رویداد است، کامل نشده باشد؛ با شکست مواجه می شود. محدودیت زمانی صرف نظر از ظرفیت سیستم اتفاق می‌افتد.

کاربردهای استاندارد سیستم بلادرنگ

از مثال های ساده ای از این نوع سیستم می توان به موارد زیر اشاره کرد:

- سیستم کنترل دیجیتال
- سیستم فرمان و کنترل
- پردازش سیگنال
- سیستم ارتباطات راه دور

از کاربردهای جدید سیستمهای بلادرنگ میتوان به شبکه‌ها اشاره کرد که در زیر مختصراً توضیح داده شده است :

شبکه بلادرنگ

افزایش تعداد کاربردهای توزیعی بلادرنگ در فراهم کردن سرویس های ارتباطی قابل پیش بینی، منفعت زیادی داشته است. با توجه به طراحی شبکه های بی سیم امکان مکان یابی در نواحی حادثه دیده را می‌دهد. یا یک شبکه کاربردی اجازه می دهد که یک جراح از راه دور ابزارآلات پزشکی را کنترل

کند. بسیاری از محققین، پروتکل شبکه های بلادرنگ و هم چنین مکانیزم سیستم عاملی توزیع داده ها را برای هر کاربردی پیش بینی می کند، توسعه داده اند.

پروتکـل انتقـال بلادرنـگ RTP - (Real-time Transport Protocol) پروتکلی بر مبنای UDP - (User Datagram Protocol) است ودر کاربردهای صوتی و تصویری استفاده می شود که در این حالت بسته های گم شده تلرانس بالا و تاخیرهای متغیر تلرانس پایین دارند. RTP همراه با RTCP - (Real Time Control Protocol) هم دسـتورات چند پخشـی ، کنترل پیام ها را که خود کیفیت فیدبک داده ها را حمل می کند، انتقال می دهد.

حتی اگر پروتکل های شـبکه بلادرنگ مورد استفاده قرار گیرند بازهم خدمات قابل پیش بینی نیستند مگراینکه سیستم عامل میزبان مقصد با مکانیزم های مناسبی برای پردازش بسته های ورودی و خروجی شبکه مجهز شده باشند. خدمات، کنترل رویدادها را به متن وقفه ها می فرستد تا به رویدادها در سطح شبکه پاسخ دهد. مثل این که یک بسته دریافت و آماده انتقال شود.

این سیستم ها هر روزه سرویس های متعدد و مفیدی را در اختیار ما قرار میدهند.به عنوان مثال:

در هنگام رانندگی ،این سیستم ها کنترل موتور و ترمز و همچنین کنترل چراغ های راهنمای رانندگی منظم را بعهده دارند

در هنگام پرواز، این سـیسـتم ها کنترل برنامه هواپیما و آگاهی از زمان فرود و بلند شـدن هواپیما ، نگهداری مسیر پرواز را بعهده دارد.

در هنگام سرماخوردگی، این سیستمها کنترل آگاهی و تنظیم وضعیت فشار خون و ضربان قلب را بعهده دارد.

در هنگام سلامت کامل، این سیستم ها ما را به وسیله بازی های الکتریکی و سواری های مفرح سرگرم میکنند.

سیستم های بلادرنگ سخت ونرم

یک رده بندی کلاسـیک، سـیسـتم های سخت یا فوری است؛ اتمام یک فرایند بعد از محدودیت زمانی مضـر اسـت که باعث ایجاد خطا در منطقه بحرانی می شود. از طرف دیگر سیستم های بلادرنگ نرم با این دیرکرد مقابله کرده و ممکن است با کیفیت بهتر پاسخ دهد.

در سـیستم های تعبیه شده، سیستم های بلادرنگ سخت در سطح پایینی از سخت افزار فیزیکی عمل می کند. برای مثال سـیسـتم کنترل موتور ماشین یک سیستم بلادرنگ سخت است چون ممکن است سـیگنال های تاخیر به موتور آسـیب برسـانند. مثال دیگر از سـیسـتم بلادرنگ سـخت، سـیستم های جاسـازی شـده در دستگاه های پزشکی مثل دستگاه تنظیم کننده ضربان قلب وپردازشگر های کنترل صنعتی.

درک قیود زمانی مخصوصاً توابع زمان/استفاده و دو فاکتوربه موقع بودن باعث شده است که بتوان تعریف دقیق تر و در عین حال عمومی تری از سیستم های بلادرنگ سخت و نرم ارائه داد.
سیستم بلادرنگ سخت نسبت به رویدادی که به محدودیت زمانی واکنش نشان می دهد، ضروری هستند به عبارتی، تعریف مهلت زمانی سخت لزوماً این نیست که این زمان غیر قابل از دست دادن باشد – بلکه مهلت زمانی سخت به سادگی تعیین می کند که یک عمل اگر مهلت زمانی اش از دست برود بی فایده است– تعمدا نتایج و پیامد های (سیستم ، درخواست ، موقعیت خاص) آن عمل نا به هنگام را مشخص نمی کند. آن نتایج و پیامدها مبناهای اصلی برای تعیین کردن معیار زمان ترتیبی لحاظ ترتیب هستند.
معمولا گارانتی های معتبر، مورد احتیاج سیستم هایی هستند که در پنجره های زمانی از خود واکنش نشان نمی دهند و باعث خسارت های بزرگی می شوند؛ مخصوصا آسیب های فیزیکی که زندگی بشر را تهدید می کند (هر چند محدودیت های زمانی گمشده باعث خطا در سیستم می شود. (سیستم هایی که محدودیت زمانی سخت دارند (به علت پتانسیل شدید خروجی محدودیت های زمانی گم شده) دارای جایگاه انرژی هستند. در زمینه سیستم های چند وظیفه ای معمولا سیاست های زمانی ارجحیت دارند (زمان بندی های اولیه.) دیگر الگوریتم های زمان بندی شامل محدودیت زمانی اولیه‌ای هستند که نادیده گرفتن سربار تعویض متن برای بار سیستم کمتر از ٪۱۰۰ کافی است.
سیستم های زمان بندی جدید مانند زمان بند قسمت بندی سازگار به مدیریت سیستم های بزرگ با ترکیبی از سیستم های بلادرنگ و غیربلادرنگ کمک می کند.
یک سیستم بلادرنگ سخت سیستمی است که فاکتور های معیار زمان ترتیبی آن :
بهینگی یک حالت باینری است که به تمام hard deadline ها در حالت بهینه و در غیر این صورت زیر بهینه برسد.
احتمال حالت بهینه قابل محاسبه باشد.
این ها تنها فاکتورهای زمان ترتیبی در بلادرنگ های سخت هستند و سیستم بلادرنگ سخت تنها همین فاکتورهای به موقع بودن را در معیارهای ترتیبی خود دارد. در فاکتورهای زمان ترتیبی فضای دو بعدی بهینگی و امکان پیشگویی بهینگی، بلادرنگ سخت در گوشه ی "ماکزیمم بهینگی / ماکزیمم امکان پیشگویی" قرار دارد. ؛ دو فاکتور در تقابل با یکدیگر نیستند. این تعریف مطابق است با تعریف متداول پژوهشگران از سیستمهای بلادرنگ سخت که یک بلادرنگ سخت سیستمی است که در آن تمام مهلت زمانی های سخت آن رسیده می شود.
لکن افراد عملگرا (کاربران، فروشندگان) از سیستم های بلادرنگ، بلادرنگ سخت و یا بلادرنگ نرم تعریفی عامیانه دارند و این اصطلاحات را به طور ناقص و غیر علمی بکار می برند.
بلادرنگ سختها عموما مانند آنچه در ادامه میآید ناشی میشوند :

تمام موجودیت های اجرا که مهلت زمانی های سـخت دارند برای تشکیل دادن سیستم بلادرنگ سخت معمولاً یک زیرسـیستم جهت تعامل با دستگاه ها با یک بلادرنگ سخت کمکی یا معمولا سیستم های غیر بلادرنگ کمکی جهت تعامل با انسانها و پایگاه داده ها دور هم جمع می شوند.

یا تمام قیود زمانی به طور مصـنوعی مجبور شـدند که مهلت زمانی های سـخت باشند به این علت که سیستم یا طراح/برنامه نویس آن نمی توانند با هیچ نوع قید زمانی دیگر سر و کار داشته باشند.

تقسـیم سـیسـتم های محاسـبه به بلادرنگ سخت مستقر در جلو و سیستم غیر بلادرنگ کمکی پشت سـیستم می تواند شیوه طبیعی و موثری در بعضی شرایط باشد. اما در بسیاری موارد اثر پخشی ابتدای محاسـبات را با جلوگیری از محدودیت زمانی سـخت و یک سـری اعمال مناسـب بلادرنگ، و یا انتهای محاسـبات با نگهداری آن از به کار گرفتن محدودیت های زمانی مشتق شده از مدیریت منابع، محدود می شود.

بـه طور اجباری محدودیت های زمانی به محدودیت های زمانی سـخت تبدیل می شـود که معمولا سازگاری و انعطاف پذیری سیستم را محدود می کند، با افزایش درخواست منابع سخت افزار قابلیت این منابع کم می شود.

سیستم های بلادرنگ نرم در دستیابی های همزمان استفاده می شود و به نگهداری تعدادی از سیستم های به روز شده با تغییر جایگاه ها نیاز دارند. برای مثال سیستم های صوتی وتصویری سیستم بلادرنگ نرم هستند. خطاهای ناشی از نتیجه محدودیت زمانی، کیفیت را تنزل می دهد اما سیستم به عمل خود ادامه می دهد.

در ترتیب دهی معیار مناسب فضای دو بعدی بهینگی یا قابل پیشبینی بهینگی، بلادرنگ نرم یک فضای کلی برای سـیستم بلادرنگ سخت است. یک سیستم می تواند به صورت سیستم بلادرنگ سخت عمل کند که قیدهای زمانی سـخت و ترتیب دهی عوامل مناسـب داشـته باشـد که همیشـه با همه قیدهای زمانی برخورد می کنند.

بعضـی از سـیستم های بلادرنگ غیرقطعی هستند. خواصی دارند که بسیار ناهماهنگ هستند که برای مدل های قابل پیشبینی احتمالی هم ناشناخته هستند. علت توالی مناسب این سیستم ها این است که درحالی که از مدل های شـبیه سـازی و دیگر مدل ها در زمینه هایی مثل هوش مصـنوعی و تئوری تصمیم استفاده می کنند، اجرا می شوند.

سـیسـتم های بلادرنگ سـخت حداقل شامل اعمالی با محدودیت زمانی سخت هستند.اما برعکس این مطلب درسـت نیسـت – سیستم های بلادرنگ نرم ممکن است اعمالی با قیدهای زمانی سخت داشته باشند. سیستم های بلادرنگ نرم، معیار مناسب ترتیب دهی سیستم بلادرنگ سخت را به کار نمی برد، بلکه معیارهای سـیسـتم بلادرنگ نرم را به کار می برد مثل "حداقل کردن تعداد قیدهای زمانی مورد انتظار"، صرف نظر از این که قیدهای زمانی سخت یا نرم هستند.

واضح است که ایجاد سیستم های بلادرنگ نرم سخت تر از ایجاد سیستم های بلادرنگ سخت هستند. بعضی از این اختلافات خیلی پیچیده تر از کاربردها و محیط های اجرایی سیستم های بلادرنگ نرم هستند اما بعضی از آ نها نتیجه موقتی دارند. هم در تئوری و هم در عمل سیستم های محاسبات بلادرنگ اصولا روی سیستم های بلادرنگ سخت متمرکز می شوند.

مهم است که به خاطر بسپاریم سیستم های بلادرنگ سخت در برابر نرم لزوما ارتباطی به طول زمان موجود ندارند. یک ماشین ممکن است اگر پردازشگر در طول ۱۵ دقیقه روشن شود زیاد گرم شود. از طرف دیگر کارت رابط شبکه اگر در کسری از ثانیه خوانده نشود ممکن است میان داده را گم کند، اما داده می تواند بدون پی آمد مضر دوباره به شبکه فرستاده شود.

سیستم بلادرنگ یکی از حالت خاص سیستم بلادرنگ سخت نیست، پس یک حالت کلی است. ترتیب عوامل بهبود یافته مناسب هر کاری ممکن است باشد –برای مثال عواملی که به طور گسترده استفاده می شوند (غیر از مجموعه محاسبات قدیمی بلادرنگ) "حداقل کردن تعداد قیدهای زمانی گمشده" و "حداقل کردن کل زمان دیرکرد"، هستند. بهینگی قابل پیش بینی غیرقطعی است و معمولا به طور اتفاقی ساخته می شوند. یک مثال رایج (غیر از مجموعه محاسبات قدیمی بلادرنگ) توالی موضوع مناسب برحسب عوامل "حداقل کردن تعداد قیدهای زمانی مورد انتظار" و "حداقل کردن جزئی از زمان دیرکرد"، هستند.

توجه کنید که بلادرنگ سخت ونرم فقط برای "سیستم" به کار می روند، زیرا تعریف آنها مبتنی بر ترتیب دهی است. به این معنی که یک سیستم امکانات مدیریت منابع است که شامل توالی (از نخ ها)، هرجا از سخت افزار و میان افزار، می باشد.

مدل های وظیفه در سیستمهای بلادرنگ

به منظور فهم عملکرد سیستم های بلادرنگ لازم است که مدل های فعالیت های بلادرنگ و مطالعه ی خصوصیت و مشخصه های آنها را گسترش دهیم.در این بخش دو مدل بنیادی وظیفه های بلادرنگ معرفی میشوند:

سیستم های بلادرنگ پریودیک و سیستم های بلادرنگ نا دوره‌ای.

وظیفه های بلادرنگ دوره ای

مثال مدل سیستم در بازه های زمانی منظم: (هر یک دقیقه یکبار) سیستم درجه حرارت را ثبت کند

در حالت عمومی یک وظیفه بلادرنگ نیازمند مقدار مشخصی منابع در طول یک دوره ی زمانی خاص است. وظیفه دوره‌ای وظیفه ای است که منابع را در واحد زمان تقاضا می کند و با تابع دورهای تناوبی نمایش داده می شود. یعنی الگوهای قطعی و مستمر وقفه ای زمانی بین درخواست های منابع دارد.

علاوه بر این نیازمندی یک وظیفه های بلادرنگ باید پردازش را توسط مهلت زمانی خاص وابسته به زمان کامل کند یعنی پیدا کردن پردازشگر (یا یک منبع دیگر).
برای ساده سازی فرض کنید که یک وظیفه بلادرنگ دوره ای درخواست ثابتی دارد(به عنوان مثال باید در هر میلی ثانیه در نشان دهنده‌ی T که (C , T) پردازشگر اجرا را شروع کند. این وظیفه ها را می‌توانیم به عنوان وظیفه های چند ارزشی توصیف کنیم بیانگر زمان سرویس که بیانگر مقدار زمان که کدام C دوره ی مورد نظر که مطابق با اختلاف زمانی بین دو درخواست موقتی است و وظیفه باید قبل از مهلت زمانی اجرا شود و کدام یک مفروض است که در شروع دوره ی درخواست بعدی اتفاق بیفتد.
به عنوان مثال یک درخواست روباتیک ممکن است شامل تعدادی وظیفه های بلادرنگ دوره‌ای (تناوبی) باشد که فعالیت هایی مانند جمع آوری اطلاعات حسگر، یا مخابره ی منظم شبکه را انجام می دهند.
فرض کنید یک رباط وظیفه ای را اجرا می کند که باید اطلاعات سنسور مادون قرمز را جمع آوری کند برای مشخص کردن این که اگر مانعی در نزدیکی و در وقفه های زمانی منظم است.
اگر پیکربندی این مادون قرمز نیازمند این باشد که هر ۵ میلی ثانیه باید جمع آوری ۲ میلی ثانیه‌ی جمع آوری و پردازش اطلاعات حسگر را کامل کند ، آنگاه وظیفه یک وظیفه تناوبی بلادرنگ است و می تواند در این مدل به صورت چندگانه توصیف شود:

(C=2ms, T=5ms)

وظیفه های بلادرنگ نامنظم

مثال مدل وظیفه بلادرنگ نامنظم –هر زمانی دما به یک مقدار مشخص شده برسد سیستم تشخیص داده و دریچه اصلی سوخت را ببندد

یک وظیفه بلادرنگ نامنظم شامل فعالیت های بلادرنگ است. این فعالیت های بلادرنگ منابع را در دوره های غیرقطعی درخواست می کنند. ممکن است هیچ مرزی وجود نداشته باشد یا تنها مرزهای آماری در ورود دوره‌ی یک درخواست واحد باشد . (نمونه‌ای از وظیفه) . هر نمونه از وظیفه همراه با یک مهلت زمانی خاص است که بیانگر زمان لازم برای کامل کردن اجراست.
نمونه های وظیفه های غیر تناوبی در سیستم های بلادرنگ با رویداد هدایت شده مانند بیرون پرتاب شدن صندلی خلبان هنگامی که فرمان به سیستم ناوبری جت داده می شود است. بسیاری از درخواست‌های عدم حساسیت زمانی همچنین در سیستم های توزیع شده نمایان می شوند که مستلزم رسانه‌ی پیوسته است. مانند:

end-host routing over a logical overlay

زمانبندی در سیستمهای بلادرنگ

یکی از بزرگترین مسئولیت های سیستم های بلادرنگ زمان بندی وظیفه ها با توجه به مهلت زمانی آنهاست با توجه به ضمانت کردن این که فعالیت های بلادرنگ به مرحله سرویس مورد نظر دست

یـافتـه‌اند. الگوریتم های زمان بندی فراوانی برای انواع مختلف وظیفه ها وجود دارد اما پایه و اســاس بسـیاری از آنها خط مشـی زمان بندی زودترین مهلت زمانی و نرخ یک نواخت اسـت. زمان بندی برای مجموع های از وظیفه ها گفته شده است که قابل اجرا و عملی باشد. همچنین یک مجموعه وظیفه قابل زمان بندی اسـت اگر یک زمان بندی قابل اجرا برای مجموعه وجود داشـته باشـد. بهره برداری با داده شـدن زمان بندی وظیفه و منبع در کسـری از زمان تخصیص یافته به فعالیت منابع در طول زمان بند فعال، .همراه است.

لازم به ذکر در اکثر مواقع هیچ ارتباط واضحی بین مسئله‌ی زمان بندی تئوری و طراحی عملی سیستم های بلادرنگ وجود ندارد.

قبل از بررســی ۲ مدل زمان بندی مرســوم RM و EDF بهتر دیدیم تا نیازهای مرســوم این نوع از سیستم ها و تعریف های اولیه زمان بندی را به تفصیل توضیح دهیم.

شناسایی نیازها

در حالت کلی می توان نیازهای سیستم های بلادرنگ را به سه دسته کلی دسته بندی کرد :

نیازهای رفتاری، نیازهای زمانی موقتی، نیاز های هزینه ای

نیازهای رفتاری

مدلی از سـیسـتم بلادرنگ که شـامل وظیفه هاست، و رفتار سیستم و منابع مورد استفاده وظیفه ها را مشـخص می کند. رفتار سـیسـتم محدودیت هایی را دراجرای آن ایجاد می کند که بر قیدهای زمان بندی اثر میگذارد. نیازهای رفتاری به طور متداول می توانند دسـتور اجرای وظیفه یا تخصـیص وظیفه باشند.

نیازهای زمانی موقتی

بیشـتر ابزارهای پیشرفته برای سیستم های بلادرنگ بر مدل کردن رفتار سیستم متمرکزند. و با فقدان نیروی بیان برای بکار بردن نیازهای زمانی مواجه اسـت. این یکی از اشکال جدی است این که سیستم های بلادرنگ در عمل رفتار موقتی دارند. رفتار موقتی بسـتگی به محیطی دارد که سـیسـتم در آن در حال فعل و انفعال است. به همین علت است که نیازها برای هر وظیفه به ندرت به طور انحصاری تعیین می شـوند و ترجیحاً برای زنجیره ی وظیفه های پشـت سر هم تابع خاص تعریف می شود؛ برای نمونه تابعی مانند cruise control ممکن است نیاز به، به روز شدن سرعت داشته باشد که به طور مستقل از تعداد وظیفه ها زمان بندی می شود.

نیاز های هزینه ای

جدا از نیازهای محض سیستم ، تمایل صنعت به بهره اقتصادی در سیستم های بلادرنگ است. به جای ساختن سخت افزار برای نرم افزار ، نرم افزار برای سخت افزار ساخته می شود. این تغییرشکل اضافی را در سیستم طراحی به علت کاهش انعطاف پذیری قرار می دهند. با این وجود انعطاف پذیری کمتر منجر به کاهش پیچیدگی زمان بندی شده و قیدهای سخت افزاری جدیدی معرفی می گردند.

زمانبندی در سیستمهای بلادرنگ

یکی از بزرگترین مسئولیت های سیستم های بلادرنگ زمان بندی وظیفه ها با توجه به مهلت زمانی آنهاست با توجه به ضمانت کردن این که فعالیت های بلادرنگ به مرحله سرویس مورد نظر دست یافته اند. الگوریتم های زمان بندی فراوانی برای انواع مختلف وظیفه ها وجود دارد اما پایه و اساس بسیاری از آنها خط مشی زمان بندی زودترین مهلت زمانی و نرخ یک نواخت است.

شناسایی نیازها

نیازهای رفتاری

- مدلی از سیستم بلادرنگ که شامل وظیفه هاست، و رفتار سیستم و منابع مورد استفاده وظیفه ها را مشخص می کند. رفتار سیستم محدودیت هایی را دراجرای آن ایجاد می کند که بر قیدهای زمان بندی اثر میگذارد. نیازهای رفتاری به طور متداول می توانند دستور اجرای وظیفه یا تخصیص وظیفه باشند

  پس نیازهای رفتاری: با رفتارهای سیستم، بشکل منظم (دوره ای) یا نامنظم و برحسب اتفاق وظایفی تعریف میشود

  مثال منظم یا دوره ای: هر یک دقیقه مقدار دمای بیرون هواپیما با ذکر زمان در سیستم ثبت کند

  مثال نامنظم (اتفاق) : در پی یک رخداد (مثلا افت فشار سیال) نوع عمل(دستور به باز شدن دریچه سوخت) بهمراه زمان، سیستم بلادرنگ اعمال و ثبت کند

نیازهای زمانی موقتی

- بیشتر ابزارهای پیشرفته برای سیستم های بلادرنگ بر مدل کردن رفتار سیستم متمرکزند. و با فقدان نیروی بیان برای بکار بردن نیازهای زمانی مواجه است. این یکی از اشکال جدی است این که سیستم های بلادرنگ در عمل رفتار موقتی دارند. رفتار موقتی بستگی به محیطی دارد که سیستم در آن در حال فعل و انفعال است. به همین علت است که نیازها برای هر وظیفه به ندرت به طور انحصاری تعیین می شوند و ترجیحاً برای زنجیره ی وظیفه های پشت سر هم

تابع خاص تعریف می شود؛ برای نمونه تابعی مانند cruise controlممکن است نیاز به، به روز شدن سرعت داشته باشد که به طور مستقل از تعداد وظیفه ها زمان بندی می شود
مثلا : در طول مدتی زمانی که هواپیما بخواهد از زمین بلند شـود در این بازه زمانی بایسـتی سـرعت حرکت لحظه به لحظه بروزرسـانی شـود و بعد از بلند شـدن دیگر به این مدل نمونه برداری از سرعت نیاز نیست

نیاز های هزینه ای

- جدا از نیازهای محض سیستم ، تمایل صنعت به بهره اقتصادی در سیستم های بلادرنگ است. به جای سـاختن سـخت افزار برای نرم افزار ، نرم افزار برای سـخت افزار ساخته می شود. این تغییرشـکل اضـافی را در سـیستم طراحی به علت کاهش انعطاف پذیری قرار می دهند. با این وجود انعطاف پذیری کمتر منجر به کاهش پیچیدگی زمان بندی شده و قیدهای سخت افزاری جدیدی معرفی می گردند
تمایل به سـود اقتصادی از سیستم ها باعث میشود نرم افزار برای سخت افزار ساخته شود مثلا نرم افزار خرید و فروش انرژی الکتریکی به کشـورهای همسـایه و تغییر مسـیر دادن کلید ها انرژی الکتریکی در راستای تغییر مقدار جریان در موقع نیاز و یا عدم نیاز

سیستم عاملهای بلادرنگ

سـیسـتم عامل های بلادرنگ ،سیستم عامل هایی چند منظوره هستند که برای کاربردهای بلادرنگ از جمله سـیسـتم های جاسـازی شـده (سـیسـتم تنظیم حرارت قابل برنامه ریزی،کنترل اسـباب های خانگی،تلفنهای موبایل)،روباتهای صنعتی، سفینه های فضایی،وسایل تحقیقات علمی ،طراحی شده اند

مشخصات سیستم عامل های بلادرنگ

1) قطعی بودن: سـیستم عامل ظرفیت کافی به تمام رخدادها در فواصل زمانی معین دارد و تمـام رخدادها با توجه به اولویت در یک دوره قابل انجام هسـتند. یعنی هر چند ممکن است سیستم عامل لختی در اجرا داشته باشد ولی سیستم عامل ظرفیت پاسخ را دارد
2) پاسخدهی خوب : سیستم عامل قطعا در زمان مقرر و مجاز پاسخ مناسب و عکس العمل لازم را انجام میدهد
3) قابل کنترل توسـط کاربر: هر چند در سـیسـتم عامل برنامه ها از قبل برنامه ریزی شده ولی با دادن حق دسترسی و مجوز به کاربر، او میتواند اولویت بندی وظیفه ها و زمان آنها را تغییر دهد

4) قابل اطمینان: به کارآمدی سیستم عامل باید کاملا اعتماد داشته باشیم و هر اشکالی در خط تولید و اتوماسیون بهیچوجه نباید ذهن ما را معطوف به ضعف و عیب در سیستم عامل معطوف کند
5) نرم در قبال خطا : یک سیستم عامل بلادرنگ بایستی اگر تشخیص داد، داده‌ها در هسته سیستم عامل تخریب شده اند نبایستی پیغام خطا بدهد و سیستم را متوقف کند بلکه بایستی اشکالات را تصحیح یا حداقل کند و به فعالیت خود ادامه دهد و با توجه به اولویت ها ادامه عملیات دهد

فلسفه طراحی این نوع سیستم عامل

- طراحی بر اساس اولویت – در این طراحی تنها زمانی وظیفه ای تعویض می شود که وظیفه ای با اولویت بالاتردرخواست دهد،به این نوع طراحی را اولویت اولیه نیز می نامند
- طراحی اشتراک زمانی – در این طراحی وظیفه بر اساس وقفه ساعت تعویض میشود که در رویدادها Round Robinنامیده می شود.

نتیجه گیری

- در زمانی که سیستم های بلادرنگ با جستجو محاسبات در زمینه های متنوع پا به عرصه گذاشت، انگیزه ای برای گسترش سیستم های موجود با مکانیزم ها و سیاست لازم برای فراهم کردن خدمات قابل پیش بینی به وجود آمد. هم چنین بسیاری از مدل کارهای سیستم بلادرنگ تنظیم تدبیر مناسب الگوریتم های زمان بندی را انجام می دهد. محققین کامپیوتر تحلیل جستجو در رو ش های جدید و درخواست ان ها در شرایطی که قابل پیش بینی و کم هزینه تر از نظر زمان باشند را ادامه خواهند داد.

مقدمه

طراحی سیستم های بلادرنگ همانند صنعت نرم افزار یک موضوع رو به افزایش در انجمنهای پژوهش های سیستم ها است درخواست های بلادرنگ و نیازهای آن ها را می توان تقریباً در محدوده ی تمام تحقیقات و پژوهش های سیستم عامل و شبکه یافت. لیست ناقص چنین دامنه ای شامل سیستم های توزیع شده ، سیستم جاسازی شده ، پردازش های پروتکل شبکه ، طراحی هواپیماها و طراحی فضاپیما ها و ..... را می توان نام برد.

استنتاج قیود

اغلب قیدهای سیستم های بلادرنگ مصنوعی هستند. و علت آن درماندگی ابزار موجود در حال حاضر برای ترجمه و تبدیل نیازهای سیستم به قیدهای مناسب است. در این تبدیل و ترجمه نیازها معمولا به چند قید ساده تقسیم می شوند. متأسفانه ، متدهای اتفاقی که برای استنتاج قیود مورد استفاده قرار می گرفتند منجر به افزایش محدودیت های زمانی و غیر اجرایی شدن سیستم می شدند. در زیر ، سعی بر آن است تا قیود در نزدیک ترین حد ممکن تا نیازها نگه داشته شود.و چگونگی مرتبط ساختن تعدادی از قیدها که به طور متناوب در سیستم های بلادرنگ ظاهر می شوند، به نیازهای سیستم نوعی بیان می شود.

مدل وظیفه

از متغیرهای زیر برای علامت گذاری و نمایش دادن ویژگی های وظیفه استفاده می کنیم.

- نماینده ی وظیفه
- Si زمان شروع فعالیت
- Ei زمان اجرای بدترین حالت
- Ri زمان واگذاری
- Diمهلت زمانی
- Piدوره
- Biزمان بلاک کردن
- Niگره های اجرایی
- همچنین C برای تعیین کردن محدودیت های اختیاری در موارد قابل اجرا استفاده می شود.

انحصاری کردن

جدا از استنتاج قیود ، باید تصمیم گرفته شود که, وظیفه ها اجازه یا عدم اجازه‌ی انحصاری کردن یکدیگر را در محدودیت های زمانی دارند یا خیر. با اجازه انحصاری کردن، امکان پیدا کردن برنامه زمانی برای طرح هایی که غیر عملی هستند به گونه ای دیگر وجود دارد.

قیود زمان بندی مطلق

این گونه محدودیت های زمانی به طور مستقیم وابسته به رفتار زمانی هستند .

زمان اجرا بیان کننده ی زمان بدترین حالت اجرا در وظیفه است. و به سخت افزاری بستگی دارد که وظیفه برای اجرا در آن زمان بندی شده است. این قابل اهمیت است که در سیستم های توزیع شده اختلاف زمان اجرای وظیفه ها مهم است. در اینجا Ei:n = زمان اجرای وظیفه استفاده شده/تعداد وظیفه ها

Ei = Ei ; N : fN = Nig

مهلت زمانی مطابق با حساسیت های مورد نیاز توسط سیستم بیان می شود. معمولاً مهلت زمانی ها به طور انحصاری برای هر وظیفه نیستند و به صورت زنجیرهای هستند. یک روش مرسوم تقسیم کردن مهلت زمانی های متناوب به چند مهلت زمانی کوچکتر است که برای هر وظیفه تعیین می شوند . این عمل ممکن است باعث اعمال فشاری غیر ضروری به مجموعه وظیفه ها شود. برای جلوگیری از این امر، مهلت زمانی های پیوسته را یک مهلت زمانی برای همه ی وظیفه ها در نظر می گیریم D را به عنوان مهلت زمانی های متناوب در نظر می گیریم و TD به عنوان تمام ,وظیفه هایی که توسط D منحصر می شوند. در نتیجه محدودیت زمانی به این صورت تعریف می شود

∧i 2 TDjDi = D:

برای اطمینان یافتن از این که هیچ محدوده زمانی ای جا نمانده است ، قیدی که در ادامه میآید نیز لازم است :

Di _ Si + Bi + Ei

زمان واگذاری که عمدتاً مربوط به پردازش های دوره ای در جایی که آنها زمان شروع را به فراخوانی وظیفه منحصر می کنند هستند. که به تفصیل توضیح داده می شود. بسیاری از الگوریتم های زمان بندی زمان واگذاری را وظیفه تحمیل می کنند به عنوان راهی برای بدست آوردن ممانعت متقابل بین وظیفه هایی که دسترسی به منبع یکسانی دارند. اشکال این تکنیک مانند اشکالی است که در بالا برای مهلت زمانی ذکر شد یعنی ریسک افزایش محدودیت زمانی در مجموعه وظیفه ها. برای حصول اطمینان از اینکه وظیفه قبل از زمان واگذاری شروع نمی شود این قید لازم است:

Si _ Ri.

Periods بیان می کند که چند بار یک وظیفه باید اجرا شود. این مربوط می شود به این که هر سیستم برای انجام فعل و انفعال با محیط چقدر باید دقیق باشد Periods ممکن است مهلت زمانی هایی را در انحصار قرار دهد. از آن جا که مهلت زمانی های یک وظیفه اغلب کمتر از periods آن است .(Di _ Pi) حتی اگر آن درست نباشد (Pi < k (Di امین فراخوانی وظیفه باید قبل ازفراخوانی k+1 ام کامل شود. یک راه بکارگیری زمان بندی دورهای برای وظیفه ، پیدا کردن کوچکترین مضرب]مشترک[تمام پریودهاست و پس از آن زمان بندی کردن هر فراخوانی به عنوان یک وظیفه واحد(انحصاری)سناریوهای مختلف این قید ها را تحمیل می کند:

Ri;k = (k − ١) _ Pi : fk;0 < Di _ Pig

Ri;k = (k − ١) _ Di : fk;0 < Di < Pig

جدایی قیدها فاصله ی مقادیر را بیان می کنند که پریود یک وظیفه باید به آن تعلق داشته باشد. در تقابل با deadline ها پریودها ممکن است توسط ماکزیمم یا/و مینیمم ارزش دوره محدود شوند که که

ازحاصل شدن بهره برداری مورد نیاز اطمینان حاصل شود ۱ و u را به ترتیب مقادیر مینیمم و ماکزیمم -ها قرار می دهیم. در نتیجه این قید به این صورت تعریف می شود:

l _ Pi_ u

قیود زمان بندی وابسته

این قیود به قیود محلی نیز مشـهورند. این قیود نشـان می دهند که چطور دو وظیفه به هم مربوط می شوند. برخی از قیود مرسوم در زیر معرفی و توضیح داده شده اند.

اولویت؛ اولویت قیدهایی هسـتند که با دسـتوری که وظیفه باید در آن اجرا شـود برخورد دارند. چنین قیدهایی غالباً از طرح سیستم مشتق می شوند .اگر وظیفه iام نسبت به وظیفه jام اولویت داشته باشد قیدی که تعریف می شود به این صورت است:

Si + Bi + Ei + c _ Sj

مسیر قیدهای مسیر متعیرهای قوی تری از قیدهای اولویت هستند. علاوه بر مقید کردن مرتبه ی اجرا آ نهـا ، کم ترین مسـافت در زمان بین دو وظیفه i و j را نیز بیان می کنند. علت این امر می تواند محدودیت ها در پردازش سـرعت محیطی که وظیفه با آن در فعل و انفعال اسـت باشـد. همچنین می تواند از تأخیر در سـخت افرار ارتباطی بین دو وظیفه ارتباطی مشـتق و حاصـل شود. اگر c مسیر باشد قید آن به صورت زیر تعریف خواهد شد:

Si + Bi + Ei + c _ Sj

تازگی قیود تازگی برعکس قیود مسـیری هستند. آن ها بیا نگر بیشترین مسیر در زمان بین دو وظیفه پشت سر هم i و j هستند این مطلب برای بیان این که یک وظیفه از نتایج حاصل شده از وظیفه دیگر وابسـته اسـت، کمک می کند. اگر دو وظیفه بیش از حد در یک زمان، دور از هم باشـند، نتیجه برای اسـتفاده دقیق نیسـت. این محدودیت به طور نمونه در کاربرد پایگاه های داده یافت می شـود. اگر قید تازگی را C بنامیم قید به این صورت تعریف خواهد شد:

Si + Bi + Ei _ Sj _ Si + Bi + Ei + c

ارتباط) همبستگی) قیود ارتباط به قیدهای تازگی وابست هاند که بیا نگر بیشترین اختلاف بین زما ن های پایان دو وظیفه همسو i وj هستند. و هنگامی ظاهر می شوند که یک وظیفه سوم از نتایج حاصل از دو وظیفه دیگراستفاده می کند و انحراف زمانی بیش از اندازه بزرگ بین پارامترها باعث ایجاد خطا در محاسبات وظیفه می شود. اگر ارتباط را با c نمایش دهیم ، قید به صورت زیر تعریف می شود.

j(Si + Bi + Ei) − (Sj + Bj + Ej)j _ c

هم آهنگی قیود هماهنگی وابسته به پریودهای دو وظیفه مرتبط است. مطلوب است که دوره مربوط به وظیفه مصـرف کننده i دقیقاً بر دوره مربوط به وظیفه تولید کننده j بخش پذیر باشد. اگر این شرایط برقرار باشـد، برای مصـرف کننده بسـیارسـاده تر است که دنباله پیغام های دریافتی را حفظ کند چون همیشه با فاصله مشابهی می رسند. این قید به صورت زیر بیان می گردد:

Pi = c _ Pj: fc0 < g

منابع

برای رسیدن به توانایی عملیات، وظیفه ها ممکن است نیازمند استفاده از اجزای سخت افزاری خاصی باشند. کارایی یا بهره وری اجزای موجود، به اینکه سخت افزار چقدر می تواند گران باشد، محدود می شود. قیود موقعیتی روی تخصیص وظیفه ها به نودهای مختلف تمرکز می کنند. یک وظیفه ممکن است نیازمند یک واحد خاص برای اجرا شدن خود باشد که در تمام نودها موجود نباشد. ما این را به سادگی با حذف نودهای غیر ممکن از مجموعه نودهای وظیفه بیان می کنیم. این یک قید سخت کد است اگر N را نودغیرممکن در نظر بگیریم به این صورت بیان می شود:

. Ni 6= N

همچنین شرایط می تواند این باشد که وظیفه های ارتباطی jوi باید در یک نود یکسان قرار گیرند. علت می تواند این باشد که سیستم عامل نیازمند این است که تمام وظیفه های در چارچوب پردازش، در یک نود یکسان قرار گیرند. علت دیگر می تواند کم کردن هزینه ها برای شبکه ارتباطی باشد. به این قیود، قیود خوشه ای می گویند. این قید به این صورت بیان می شود:

Ni = Nj

قیود موقعیتی اغلب توصیه های اجرایی هستند که توسط طراح مطرح می شوند و کمتر به طور متناوب، داده و تصریح می شوند.

ارتباط میان وظیفه ها نیازمند رسانه ارتباطی است. زمان فرستادن پیغام بستگی به کارایی پیشنهادی توسط کانال ارتباطی دارد. پیغام ها بین وظیفه ها فرستاده می شوند که باید در شبکه ی ارتباطی ای که بر زمان بندی وظیفه ها اثر می گذارد تنظیم زمانی شوند. زمان بندی پیغام ها non preemptive است. ساختارهای قیدی از پیش تعریف شده ای برای این نوع زمان بندی ها وجود دارد مانند قیدهای پشت سر هم.

ORing

قیود o ring آنچنان مرسوم نیستند اما می توانند برای بیان عملیات متناوب مورد استفاده قرار گیرند. به عنوان مثال، وظیفه i توسط وظیفه دیگرj حتما نباید قبضه شود (رابطه پیشگیری)، به این معنی است که وظیفه i باید قبل یا بعد از وظیفه j اجرا شود. هرگونه قید می تواند ored باشد. برای c قید به راحتی تعریف می شود:

constraint1 _ constraint2 _ :::_ constraintc

تخصیص حافظه و زمان بندی

بسیاری از الگوریتم های زمان بندی منتشر شده بلادرنگ نوع hard-coded است و به نسبت محدودیت ها، ویژگی های وظیفه ومنابع، می توانند به کار روند. هرچند در یک متد خاص در اغلب شرایط، تجاوز

به عمل کردهای عمومی ساختار را برای داشتن خصوصیات جدید پیچیده می کند. به همین علت است که روند برنامه نویسی قیود را انتخاب کردیم. ابزاری که برای آزمایش ها انتخاب می کنیم SICStus Prolog است و حل کننده ی قید همراه آن برای دامن ههای محدود نیازمندی های یک سیستم بلادرنگ شناخته شده می تواند با استفاده از قیود اولیه که با قیود پیچیده ترکیب می شوند بیان شود. مزیت این عمومیت این است که معرفی قیود پیچیده جدید تا زمانی که از ترکیب قیود ساده ایجاد شوند بسیار ساده خواهد بود. ترجمه این قیدها به کد بسیار صریح و راحت است. یک خصوصیت جالب امکان راهنمایی جستوجو برای زمان بندی عملی توسط تغییر دادن پارامترهای جستوجو است. با استفاده از پارامترهای صحیح زمان جستوجو می تواند به اندازه قابل توجهی کاهش باید. امیدواریم که امکان پذیر باشد که به صورت خودکارمشخص شود که کدام کشف کننده (heuristic) مناسب ترین برای یک نمونه مثال خاص است.

سیستم عاملهای بلادرنگ

سیستم عامل های بلادرنگ ،سیستم عامل هایی چند منظوره هستند که برای کاربردهای بلادرنگ از جمله سیستم های جاسازی شده (سیستم تنظیم حرارت قابل برنامه ریزی،کنترل اسباب های خانگی،تلفنهای موبایل)،روباتهای صنعتی، سفینه های فضایی،وسایل تحقیقات علمی ،طراحی شده اند. سیستم عامل های بلادرنگ کمک شایانی در سهولت ساخت سیستم های بلادرنگ کردند اما ضمانت قطعی در بلادرنگ بودن جواب نهایی آنها نداشتند: بلکه این نیاز باید در نرم افزارهای مربوط رعایت شود.سیستم عامل های بلادرنگ نیازی ضروری به داشتن توان عملیاتی بالایی ندارند بلکه بیشتر،امکاناتی را فراهم می سازند، که در صورت استفاده به جا و درست از آنها، ضمانت کننده مهلت زمانی است که عموماً)در بلادرنگ های نرم افزاری(و قطعاً)در بلادرنگ های سخت افزاری(یافت می شود.سیستم عامل های بلادرنگ به طور مرسوم،از الگوریتم های زمان بندی شده خاصی،جهت تأمین توسعه دهندگان بلادرنگ با وسایلی استفاده می کند.این وسایل برای قطعیت رفتار در سیستم های نهایی ضروری هستند،.ارزش سیستم عامل های بلادرنگ بیشتر در روش های سریع و قابل پیش بینی شده که پاسخگوی رخدادهای خاص هستند، نهفته است تا توان عملیاتی آنها.بنابراین از فاکتورهای اساسی در سیستم عامل های بلادرنگ می توان به حداقل تأخیر در وقفه ها و حداقل تأخیر در تعویض نخ ها اشاره کرد.

نمونه های اولیه و بزرگ این نوع سیستم عامل ها که اصطلاحاً "برنامه های آنترلی " نامیده می شوند،برای سیستم خطوط هوایی Sabre توسط IBM و خطوط هوایی آمریکا طراحی و توسعه یافت.

فلسفه طراحی این نوع سیستم عامل

دو نوع طراحی پایهای در این زمینه وجود دارد:

طراحی بر اساس اولویت

در این طراحی تنها زمانی وظیفه ای تعویض می شـود که وظیفه ای با اولویت بالاتردرخواسـت دهد،به این نوع طراحی را اولویت اولیه نیز می نامند.

طراحی اشـتراک زمـانی – در این طراحی وظیفـه بر اسـاس وقفه سـاعت تعویض میشـود که در رویدادها Round Robinنامیده می شود.

طراحی سیستم های اشتراک زمانی بیشتر بر اساس تکرر تعویض متن است تا نیاز واقعی آنها به تعویض. اما در عوض، سـیسـتم های چند منظوره قطعی تر و هموارتری را ایجاد می کنند که باعث می شـود چنین تصوری ایجاد شود که هر فرآیند از کل منابع ماشین استفاده می کند.

در طراحی CPU های پیشین cycleهای زیادی برای تعویض)تغییر) بین وظایف مختلف نیاز بود که در این cycleها CPU قادر به انجام کار خاصی نبود. لذا سیستم عامل های پیشین با کاهش تعداد تعویض های وظایف سعی در کم کردن زمان تلف شده درCPUها داشتند.

CPUهای جدیدتر زمان بسیار کمتری را برای تعویض بین وظایف صرف می کنند. در بهترین حالت می توان به پردازنده های پرسرعت اشاره داشت که برای تعویض بین وظایف CYCLEای را صرف نمی کند. سـیستم عامل های بلادرنگ تقریباً همگی سیستم اشتراک زمانی را با سیستم الویت زمانی پیاده سازی کرده اند.

مشخصات سیستم عامل های بلادرنگ

سیستم عامل های بلادرنگ را می توان با داشتن ملزومات یگانه در پنج حوزه عمومی زیر،مشخص نمود:

قطعی بودن

پاسخدهی

کنترل کاربر

قابلیت اطمینان

نرمش با خطا

سـیستم عاملی قطعی است که عملیات خود را در زمان های ثابت یا فواصل زمانی از پیش تعیین شده انجام دهد. وقتی چند فرآیند در رقابت برای منابع و زمان پردازنده هسـتند ، هیچ سیستمی نمی تواند قطعی باشـد.در یک سیستم عامل بلادرنگ ، درخواست های فرآیند برای خدمت توسط رخدادها،اولاً به سـرعتی که می تواند به وقفه ها پاسـخ دهد و ثانیاً به اینکه آیا سـیسـتم ظرفیت کافی برای اداره تمام درخواست ها،در زمان معلوم را دارد یا خیر،وابسته است.

یک معیار مفید برای قبلیت عملکرد قطعی سیستم عامل،حداکثر تأخیر از زمان ورود یک وقفه دستگاه با اولویت بالا،تا زمان شروع خدمت است. در سیستم عامل های غیر بلادرنگ این تأخیر ممکن است در

محدوده ده ها تا صـدها میلی ثانیه باشـد،در حالی که در یک سـیسـتم عامل بلادرنگ ممکن است این تأخیر حد بالایی از حدود چند میکرو ثانیه تا یک میلی ثانیه باشد. یک مشخصه مربوط ولی مجزا،پاسخ دهی است.قطعی بودن درباره این است که سیستم عامل قبل از تصدیق یک وقفه چه مفدار تأخیر دارد. پاسـخ دهی مربوط به این اسـت که یک سـیستم عامل پس از تصدیق،چه مدت صرف خدمت دادن به وقفه می نماید.

قطعی بودن و پاسخ دهی به همراه هم،زمان پاسخ به رخدادهای خارجی را تعیین می کنند. ویژگی زمان پاسـخ در سیستم های بلادرنگ بسیار حساس است،زیرا چنین سیستم هایی باید نیازهای زمانی اعمال شده توسط افراد،دستگاه ها و جریان داده ها در خارج از سیستم را رعایت کنند.

عموماً کنترل کاربر در یک سـیسـتم بلادرنگ بسـیار وسیع تر از کنترل کاربرد در سیستم عامل عادی است. در سیستم عامل های عادی کاربر یا هیچ گونه کنترلی بر عمل زمان بندی ندارد رهنمودهای کلی ارائه کند.ولی در یک سـیستم بلادرنگ لازم است به کاربراجازه کنترل دقیق اولویت وظیفه داده شود و بتواند میان وظیفه های سخت و نرم تفاوت قائل شود.

قابلیت اطمینان نوعاً در سـیسـتم های بلادرنگ بسـیار مهم تر از سیستم های عادی است. یک خرابی کذرا در سیستم غیر بلادرنگ ممکن است تا تعمیر یا تعویض،منجر به سطح خدمت پایین تر گردد.ولی در سـیسـتم بلادرنگ که در حال پاسـخ دهی و کنترل رخدادها در زمان حقیقی است،از دست رفتن یا کاهش کارآمدی یک پردازنده می تواند عواقب فاجعه آمیزی داشته باشد.

نرمش خطا،به مشخصه ای اشاره دارد که با خرابی سیستم،تا حد ممکن قابلیت ها و داده های آن حفظ شـود.مثلاً یک سـیسـتم unix سـنتی،وقتی خراب شـدن داده ها در هسـته سـیستم عامل را تشخیص دهد،یک پیام شـکسـت بر روی میز فرمان متصدی ارائه کرده، محتویات حافظه را برای تجزیه و تحلیل بعدی شـکسـت،بر روی دیسک تخلیه می کند و به اجرای سیستم پایان می دهد.در مقابل یک سیستم بلادرنگ سعی بر این دارد که اشکال را تصحیح گند یا در حالی که به اجرا ادامه می دهد تأثیرات اشکال را حداقل سازد. یکی از موارد مهم نرمش خطا به عنوان پایداری شناخته می شود. یک سیستم بلادرنگ پایدار در مواردی که ارضای تمام مهلت – های زمانی وظیفه غیر ممکن باشد ، مهلت های زمانی وظیفه های بسیار حساس و اولویت بالاتر را برآورده می کند.

نتیجه گیری

در زمانی که سیستم های بلادرنگ با جستجو محاسبات در زمینه های متنوع پا به عرصه گذاشت، انگیزه ای برای گسترش سیستم های موجود با مکانیزم ها و سیاست لازم برای فراهم کردن خدمات قابل پیش بینی به وجود آمد. هم چنین بسیاری از مدل کارهای سیستم بلادرنگ تنظیم تدبیر مناسب الگوریتم های زمان بندی را انجام می دهد. محققین کامپیوتر تحلیل جستجو در رو ش های جدید و درخواست ان ها در شرایطی که قابل پیش بینی و کم هزینه تر از نظر زمان باشند را ادامه خواهند داد. سیستم های بلادرنگ زمینه گسترده ای برای مطالعات بیشتر است و در سال های آینده پیشرفت زیادی خواهد کرد.

****

زمانبندی بلادرنگ : Real-time Scheduling

سیستم بلادرنگ (Real time) سیستمی است که در آن زمان پاسخگویی به وقایع خیلی اهمیت دارد. به عنوان مثال یک نیروگاه اتمی را در نطر بگیرید ، برخی کمیتها باید تحت کنترل دقیق باشند مثلا در اثر پرتاپ نوترونها به اتمها ، نوترونهای جدیدی آزاد می شوند و نوترونهای آزاد شده به اتمهای دیگر برخورد می کند و نوترونهای جدید آزاد میشود و بهمین ترتیب . اگر تعداد نوترونهای آزاد شده از یک حدی بیشتر باشد انفجار نوترونی اتفاق می افتد.پس غلظت نوترونها باید تحت کنترل دقیق باشد . حتی یک ثانیه بعد یا یک دقیقه یا یک ساعت بعد از انفجار اگر پاسخ دهد ، هیچ ارزشی ندارد.

سیستم مانیتورینگ بخش I.C.U یک بیمارستان یک مثال دیگر از سیستم Rea-time است.

سیستمهای Real-time به دو دسته تقسیم می شوند :

بلادرنگ سخت Hard Real-time

بلادرنگ نرم Soft Real-time

بلادرنگ سخت (Hard Real-time) سیستمی است که در یک مهلت زمانی یا پاسخ میدهد یا هیچ ، مانند مثالهای فوق.

سیستم بلادرنگ نرم سیستمی است که در بعضی از مواقع آماده نشدن پاسخ در مهلت زمانی تعیین شده قابل تحمل است مانند سیستم پخش یک قطعه موسیقی از روی CD یا پخش یک Video-Clip از روی یک VCD .

- در یک سیستم بلادرنگ وقایعی رخ می دهد ؛ برنامه به تعدادی پروسس تقسیم می شود و هر پروسس برای پاسخگویی به یک نوع واقعه است.

وقایع در یک سیستم بلادرنگ به دو دسته تقسیم می شوند:

متناوب Periodic

غیر متناوب Aperiodic

وقایع متناوب با دوره تناوب مشخص تکرار می شوند .

وقایع غیر متناوب به صورت تصادفی رخ می دهند (زمان رخ داد مشخصی ندارند)

از آنجا که پردازش مربوط به هر واقعه بخشی از زمان CPU را اشغال می کند ، ممکن است پاسخ کلیه وقایع در مهلت مشخص امکان پذیر نباشد . مخصوصا اگر قدرت پردازش بالا نباشد . فرض کنید وقایع متناوب عبارتند از :

E مخفف Event است

E1
E2
.
.
.
Em

اگر فرض کنیم process مربوط به واقعه i ، Ci ثانیه وقت CPU را اشغال می کند؛ آنگاه :

CiFi= زمان اشغال شده در ثانیه از وقت CPU برای واقعه i

و کل زمان صرف شده از CPU در یک ثانیه برای پاسخگویی به وقایع عبارتست از :

I =Sigma CiFi = C1F1+C2F2+…..CmFm

بدیهی است اگر مقدار I کوچکتر یا مساوی یک ثانیه باشد یعنی قدرت پردازش سیستم برای پلسخگویی به کلیه وقایع کافی است . لذا شرط اینکه سیستم قادر به پاسخگویی به کلیه وقایع باشد این است که

Sigma CiFi <= 1
Sigma Ci /Pi <=1

در این صورت می گویند سیستم قابل زمانبندی (Schedulable) است .

(از زمان Process Switch صرفنظر شده .)

مثال : فرض کنید یک سیستم بلادرنگ از سه واقعه متناوب با دوره های تناوب 100، 200 ،500 میلی ثانیه

تشکیل شده است .اگر هر واقعه به ترتیب به 50 و 30و100 میلی ثانیه زمان CPU نیاز داشته باشد،

(الف) آیا سیستم فایل زمانبندی (Schedulable) است ؟

Sigma Ci / Pi = c1/p1 + c2/P2 +C3/P3=50/100+30/200+100/500= 0.5+0.15+0.2=0.85<=1

لذا قابل زمانبندی است

(ب) اگر واقعه چهارمی با دوره تناوب 1 ثانیه اضافه شود آیا سیستم هنوز قابل زمانبندی است؟

Sigma Ci / Pi = c1/p1 +..+C4/P4=0.85 + C4/1<=1

C4<=0.15ثانیه

C4<=150ms

لذا فقط در صـورتیکه Process مربوطه به P4 حداکثر به 150 میلی ثانیه زمان CPU نیاز داشـته باشد ، سیستم قابل زمانبندی است.

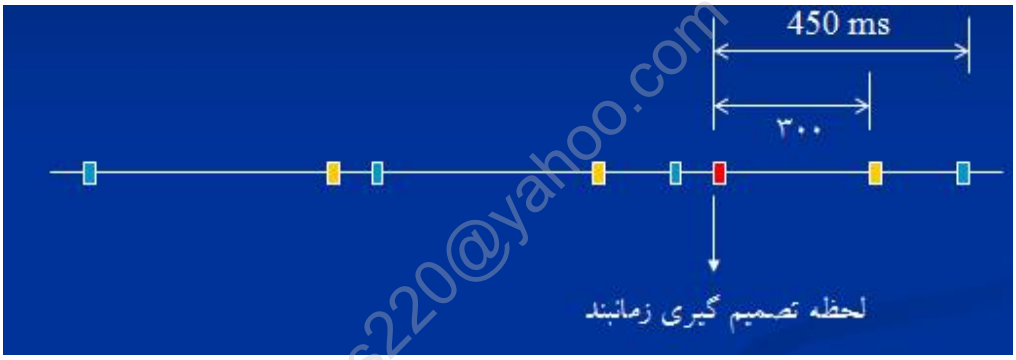
برخی از الگوریتمهای زمانبندی بلادرنگ :

1- الگوریتم نرخ یکنواخت (Rate Monotonic Algorithm) :

یک الگوریتم دارای اولویت اسـت که به هر پروسـس ، اولویتی متناسب با فرکانس آن رخداد اختصاص داده شود . به عنوان مثال اگر Process 1 ، دارای دوره تناوب 20 میلی ثانیه است و به آن اولویت 50 داده می شود و به Process2، دارای دوره تناوب 100 میلی ثانیه است اولویت 10 داده می شود

2- الگوریتم ابتدا زودترین مهلت (Earliest Deadline First)

مهلت برای یک واقعه متناوب برابر زمان رخداد واقعه بعدی خواهد بود . مثال :

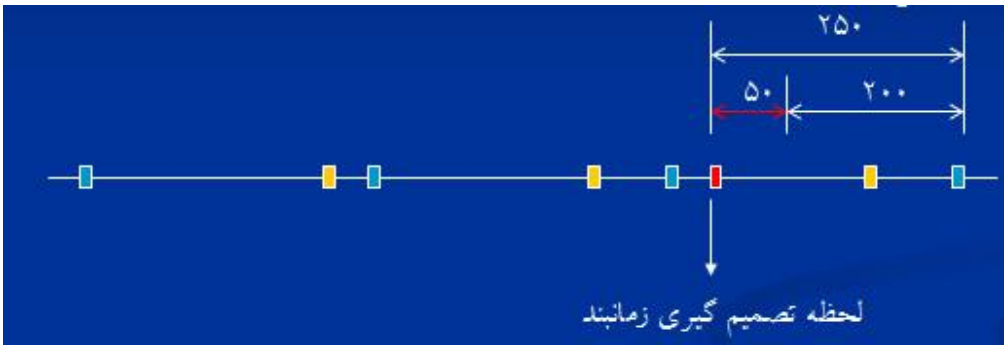


این الگوریتم می گوید که لیستی آماده اجرا داشته باشیم که در آن پروسسها به ترتیب مهلتشان Sort شده اند.

سپس پروسسی CPU میگیرد که اول صف باشد یعنی پروسسی که کمترین مهلت را دارد (فرصتش ار همه کمتر است).

الگوریتم کمترین لختی (Least Laxity) :

تعریف مقدار لختی یک پروسـس :حداکثر مقدار زمانی که پروسـس می تواند در آن مدت آماده باقی بماند و اجرا نشود .مثال : اگر یک پروسسی 200 میلی ثانیه وقت CPU نیاز داشته باشد و 250 میلی ثانیه مهلت داشته باشد و لختی آن 50 میلی ثانیه است .



الگوریتم کمرین لختی (Least Laxity) می گوید پروسسی انتخاب شود که کمترین لختی را دارد

برخی از مشـخصات سیستمهای عامل بلادرنگ عبارتند ار : کوچکی اندازه ، وقفه زمانی سریع ، تعویض متن سریع ، کوتاه بودن فواصل زمانی از کار انداختن وقفه ها

برخی از مشـخصات سیستمهای عامل بلادرنگ عبارتند ار : کوچکی اندازه ، وقفه زمانی سریع ، تعویض متن سریع ، کوتاه بودن فواصل زمانی از کار انداختن وقفه ها

****

OSI

در صـنعت، کامپیوترهای و سرورهای کامپیوتری متفاوتی برای اتوماسیون بصورت شبکه متصل هستند که برای درک مفهوم شبکه بایستی مدل هفتگانه OSI را شناخت

شاید برای شما هم این مسئله پیش آمده باشد که مدل مرجع OSI چیست و چگونه به برقراری ارتباط نرم‌افزارها در شبکه کمک میکند ؟

بعنوان مثال تصور کنید که نرم افزاری قرار است که تحت شبکه به سیستم دیگری متصل شود و تبادل اطلاعات داشته باشد. کسی که نرم افزار تحت شبکه را می نویسد برای شبکه درایور نمی نویسد ، فقط به شکلی نرم افزار خود را مینویسد که سیستم عامل بتواند تحت شبکه موارد منتقل شده را درک کند . سازنده کارت شبکه برای کارت شبکه خود درایوری را ارائه می دهد که به وسیله آن این سیستم عامل کارت شـبکه را براحتی تشـخیص و با آن ارتباط برقرار می کند . این سـیسـتم عامل ویندوز اسـت که قابلیت ارسـال داده ها بر روی کارت شـبکه و سپس خود شبکه را به شکلی ایجاد می کند که اطلاعات نرم افزار بتواند براحتی از طریق شبکه منتقل شوند .

به یاد داشته باشید که کارت شبکه یا همان NIC فقط برای ارسال و دریافت داده ها ساخته شده است . این کـارت هیچ چیز در مورد ویـنـدوز و نرم افزار های کاربردی آن نمیداند و حتی در مورد پروتکل ها هم هیچ تصوری ندارد ، در اصل یک دستگاه کاملا Passive محسوب می شود . در مثال فوق ، در واقع سـه لایـه کـاری وجود دارد ، لایـه نرم افزارهـای کـاربردی (Application) ، لایـه سـیسـتم عامل(Operating System) و لایه فیزیکی سـخت افزار . تمامی این لایه های وجود دارند اما نه دقیقا با همین عنوان ، هر کدام از این لایه های به خودی خود به یک سری زیر لایه تقسیم می شوند .

مدل شبکه ای که ویندوز و بسیاری دیگر از سیستم عامل های دیگر از آن استفاده می کنند به نام مدل مرجع OSI شـناخته می شود. واژه OSI مخفف کلمه Open System Interconnection است .

مدل OSI شـامل هفت لایه مختلف اسـت که هر کدام از لایه های موجود در این مدل مرجع وظیفه خاصـی را بر عهده دارند و کار خاصـی بر عهده هر کدام از این لایه ها می باشـد . این لایه ها بین لایه بالاتر و پایینتر خود قرار گرفته و به آنها سرویس می دهند . در واقع هر لایه با لایه پایینتر و بالاتر خود وابسته است .

پشـت سر هم بودن و نظم بسته های اطلاعاتی یا packet ها در شبکه بسیار مهم است به دلیل اینکه هر پروتکلی برای خود حداکثر اندازه‌ای برای بسته اطلاعاتی تعیین کرده است. برخی اوقات اندازه بسته اطلاعاتی از اندازه تعیین شـده آن بیشـتر می شـود و به همین دلیل داده ها به بسـته های کوچکتری تقسـیم می شوند و هر یک تشکیل یک بسته را می دهند به اینکار به اصطلاح Fragment کردن می گویند .

هفت لایه OSI (Open Systems Interconnection)

این مدل در سال 1984 توسط ISO توسط یک سازمان بین المللی استاندارد سازی ارائه گردید.در مدل OSI از هفت لایه برای تشریح فرآیندهای مربوط به ارتباطات استفاده می گردد. هریک از لایه ها مسئولیت انجام عملیات خاصی را برعهده دارند، مدل OSI به عنوان یک مرجع و راهنما برای شناخت عملیات مربوط به ارتباطات استفاده می گردد، به منظور آشنا یی با نحوه عملکرد یک شبکه ، مطالعه مدل OSI، مفید می باشد. توسط OSI می توانید چگونگی انتقال اطلاعات میان دو نرم افزار بر روی دو کامپیوتر را مشاهده کنید

لایه کاربردی یاApplication Layer

لایه ی 7 (Application): این لایه با سیستم عامل و یا برنامه های کاربردی ارتباط دارد. کاربران با استفاده از نرم افزارهای کاربردی متفاوت قادر به انجام عملیات مرتبط با شبکه خواهند بود. مثلا کاربران می توانند اقدام به ارسال فایل خواندن پیام ارسال پیام و ... نمایند.

بالاترین لایه در مدل مرجع OSI لایه کاربرد یا Application است . اولین نکته ای که در خصوص لایه کاربردی یا Application باید بدانید این است که به هیچ عنوان این لایه با نرم افزارهای کاربردی ارتباطی ندارد و صرفا یک تشابه اسمی است .در عوض این لایه محیطی را ایجاد میکند که نرم افزارهای کاربردی بتوانند از طریق آن با شبکه ارتباط برقرار کنند . برای اینکه درک بهتری از لایه کاربرد داشته باشید فرض کنید که یک کاربر با استفاده از نرم افزار Internet Explorer قصد دارد از طریق پروتکل FTP یک فایل را در شبکه منتقل کند . در این مورد لایه کاربرد به وظیفه برقراری ارتباط با پروتکل FTP برای انتقال فایل را بر عهده دارد . این پروتکل بصورت مستقیم برای کاربران قابل دسترسی نیست ، کاربر بایستی با استفاده از یک نرم افزار رابط مانند Internet Explorer برای برقراری ارتباط با پروتکل مورد نظر استفاده کند . **بصورت خلاصه وظیفه لایه کاربردی، رابط بین کاربر و شبکه است** و تنها قسمتی از این مدل هفت لایه ای است که کاربر تا حدی می تواند با آن ارتباط برقرار کند .

لایه نمایش یاPresentation

لایه ی 6 (Presentation): این لایه داده های مورد نظر خود را از لایه Application اخذ و آنها را بگونه ای تبدیل خواهد کرد که توسط سایر لایه ها قابل استفاده باشد.

فعالیت لایه نمایش یا Presentation تا حدی پیچیده است اما همه کارهایی که این لایه انجام می دهد را میتوان در یک جمله خلاصه کرد ، لایه نمایش اطلاعات را از لایه کاربرد دریافت میکند و در قالبی در می آورد که برای لایه های پایینتر قابل درک باشد . همچنین برعکس این عمل را نیز انجام

میدهد یعنی زمانی که اطلاعاتی از لایه نشست یا Session به این لایه وارد می شود ، این اطلاعات را به گونه ای تبدیل می کند که لایه کاربرد بتواند آنها را درک کرده و متوجه شود . دلیل اهمیت این لایه این است که نرم افزارهای اطلاعات را به شیوه ها و اشکال مختلفی نسبت به یکدیگر بر روی شبکه ارسال می کنند . برای اینکه ارتباطات در سطح شبکه ها بتوانند برقرار شوند و به درستی برقرار شوند شما بایستی اطلاعات را به گونه ای ساختاردهی کنید که برای همه انواع شبکه ها استاندارد و قابل فهم باشد. بطور خلاصه **وظیفه اصلی لایه نمایش، قالب بندی اطلاعات یا Formatting اطلاعات است** .
معمولا فعالیت هایی نظیر رمزنگاری و فشرده سازی از وظایف اصلی این لایه محسوب می شود .

لایه نشست یا جلسه یاSession

لایه ی 5 (Session) :این لایه مسئول ایجاد ، پشتیبانی و ارتباطات مربوطه با دستگاه دریافت کننده اطلاعات است

وقتی داده ها به شکلی قابل درک برای ارسال توسط شبکه در آمدند ، ماشین ارسال کننده بایستی یک Session با ماشین مقصد ایجاد کند . منظور از Session دقیقا شبیه ارتباطی است که از طریق تلفن انجام می شود ، شما برای ارسال اطلاعات از طریق تلفن حتما بایستی با شخص مورد نظرتان تماس برقرار کنید .اینجا زمانی است که لایه نشست وارد کار می شود ، **لایه نشست، وظیفه ایجاد ، مدیریت و نگهداری و در نهایت خاتمه یک Session را با کامپیوتر مقصد بر عهده دارد** .نکته جالب در خصوص لایه نشست این است که بیشتر با لایه کاربرد مرتبط است تا لایه فیزیکی ، شاید فکر کنید که بیشتر Session ها بین سخت افزارهای و از طریق لینک های شبکه ایجاد می شوند اما در اصل این نرم افزارهای کاربردی هستند که برای خود Session با نرم افزار مقصد ایجاد میکنند . اگر کاربری از تعدادی نرم افزار کاربردی استفاده میکند ، هر کدام از این نرم افزاها به خودی خود می توانند یک Sessionبا نرم افزار مقصد خود برقرار کنند که هر کدام از این Session ها برای خود یک سری منابع منحصر به فرد دارد .

لایه انتقال یا Transport

لایه ی 4 (Transport) :این لایه مسئول پشتیبانی کنترل جریان داده ها و و بررسی خطاء و بازیابی اطلاعات بین دستگاه های متفاوت است . کنترل جریان داده ها ، بدین معنی است که لایه فوق در صورتی که اطلاعاتی از چندین برنامه ارسال شده باشد ، داده های مربوطه به هر برنامه را به یک streamآماده تبدیل تا در اختیار شبکه فیزیکی قرار داده شوند.

لایه انتقال وظیفه نگهداری و کنترل ریزش اطلاعات یا Flow Control را بر عهده دارد . اگر به خاطر داشته باشید سیستم عامل ویندوز به شما این اجازه را میدهد که همزمان از چندین نرم افزار استفاده کنید . خوب همین کار در شبکه نیز ممکن است رخ بدهد ، چندین نرم افزار بر روی سیستم عامل تصمیم میگیرند که بصورت همزمان بر روی شبکه اطلاعات خود را منتقل کنند . لایه انتقال اطلاعات

مربوط به هر نرم افزار در سیستم عامل را دریافت و آنها را در قالب یک رشته تکی در می آورد . همچنین این لایه وظیفه کنترل خطا و همچنین تصحیح خطا در هنگام ارسال اطلاعات بر روی شبکه را نیز بر عهده دارد . بصورت خلاصه **وظیفه لایه انتقال این است که از رسیدن درست اطلاعات از مبدا به مقصد اطمینان حاصل کند** ، انواع پروتکل های اتصال گرا یا Connection Oriented و غیر اتصال گرا Connection Less در این لایه فعالیت میکنند .

لایه شبکه یاNetwork

لایه ی 3 (Network) :در این لایه روش ارسال داده ها برای دستگاه گیرنده تعیین خواهد شد. پروتکل های منطقی ، روتینگ و آدرس دهی در این لایه انجام خواهد شد.

وظیفه لایه شبکه این است که چگونگی رسیدن داده ها به مقصد را تعیین کند . این لایه وظایفی از قبیل آدرس دهی ، مسیریابی و پروتکل های منطقی را عهده دار است . بیشتر مخاطبین این مقاله افراد تازه وارد به دنیای شبکه هستند پس من هم وارد جزئیات فنی این لایه نمی شوم ، اما به شما می گویم که لایه شبکه مسیرهای منطقی یا Logical Path بین مبدا و مقصد ایجاد میکند که به اصطلاح مدارهای مجازی یا Virtual Circuits نامگذاری می شوند ، این مدارها باعث میشوند که هر بسته اطلاعاتی بتواند راهی برای رسیدن به مقصدش پیدا کند . لایه شبکه همچنین وظیفه مدیریت خطا در لایه خود ، ترتیب دهی بسته های اطلاعاتی و کنترل ازدهام را نیز بر عهده دارد .ترتیب بسته های اطلاعاتی بسیار مهم است زیرا هر پروتکلی برای خود یک حداکثر اندازه بسته اطلاعاتی تعریف کرده است . برخی اوقات پیش می آید که بسته های اطلاعاتی از این حجم تعریف شده بیشتر می شوند و به ناچار اینگونه بسته های به بسته های کوچکتری تقسیم می شوند و برای هر کدام از این بسته های اطلاعاتی یک نوبت یا Sequenceداده می شود که معلوم شود کدام بسته اول است و کدام بسته دوم و ....به این عدد به اصطلاح Sequence Number هم گفته می شود .

وقتی بسته های اطلاعاتی در مقصد دریافت شدند ، در لایه شبکه این Sequence Number ها چک میشود و به وسیله همین Sequence Number است که اطلاعات به حالت اولیه باز میگردند و تبدیل به اطلاعات اولیه می شوند . در صورتیکه یکی از این بسته های به درستی دریافت نشود در همان لایه شبکه از طریق چک کردن این عدد مشخص می شود که کدام بسته اطلاعاتی دریافت نشده است و طبیعتا مجددا در خواست داده می شود .

اگر درک این مطلب کمی برایتان دشوار است تصور کنید که قرار است برای یکی از دوستانتان یک نامه بلند بالا بنویسید ، اما کاغذ به اندازه کافی ظرفیت برای درج مطالب شما ندارد ، خوب شما میتوانید چندین کاغذ را استفاده کرده و متن خود را در قالب چندین نامه بنویسید و برای هر کدام از صفحات نامه شماره صفحه بگذارید، بعد از اینکه نامه در مقصد به دوستتان رسید با استفاده از شماره صفحه ها

میتواند نامه را به ترتیب خوانده و اطلاعات را به درستی دریافت کند . این دقیقا همان کاری است که لایه شبکه انجام می دهد .

لایه انتقال داده یاData Link

لایه ی 2 (Data) :این لایه ، پروتکل های فیزیکی به داده اضافه خواهند شد. در این لایه نوع شبکه و وضعیت بسته های اطلاعاتی (Packet) نیز تعیین می گردند.

لایه انتقال به خودی خود به دو زیر لایه به نام های MAC که مخفف Media Access Control و LLC که مخفف Logical Link Control هست تقسیم می شود . زیر لایه MAC همانطوری که از نامش پیداست **شناسه سخت افزاری کامپیوتر که در واقع همان آدرس MAC کارت شبکه است را به شبکه معرفی میکند** . آدری MAC آدرس سخت افزاری است که در هنگام ساخت کارت شبکه از طرف شرکت سازنده بر روی کارت شبکه قرار داده می شود و در حقیقت Hard Code می شود .این در حقیقت مهمترین فاکتوری است در آدرس دهی که کامپیوتری از طریق آن بسته های اطلاعاتی را دریافت و ارسال می کند . زیر لایه LLC وظیفه کنترل Frame Synchronization یا یکپارچه سازی فریم ها و همچنین خطایابی در لایه دوم را بر عهده دارد .

لایه فیزیکی یاPhysical Layer

لایه ی 1 (physical) : این لایه در ارتباط مستقیم با سخت افزار بوده و خصایص فیزیکی شبکه نظیر : اتصالات ، ولتاژ و زمان را مشخص می نماید.

مدل OSI بصورت یک مرجع بوده و پروتکل های پشته ای یک و یا چندین لایه از مدل فوق را ترکیب و در یک لایه پیاده سازی می نمایند.

لایه فیزیکی در حقیقت **به ویژگیهای سخت افزاری کارت شبکه اشاره می کند** . لایه فیزیکی به مواردی از قبیل زمانبندی و ولتاژ برقی که قرار است در رسانه منتقل شود اشاره می کند . این لایه در واقع تعیین میکند که ما به چه شکل قرار است اطلاعات خود را و از طریق چه رسانه ای انتقال دهیم ، برای مثال رسانه ما کابل فلزی است یا تجهیزات بی سیم ؟ برای راحت کردن درک این لایه بعتر است بگوییم لایه فیزیکی تعیین میکند که اطلاعات چگونه در سطح رسانه دریافت و ارسال شوند . عملیات Codingنیز که به معنای تعیین کردن صفر و یک در رسانه است نیز در این لایه انجام می شود .

کارکرد دو طرفه

در فوق ذکر شد که در مدل مرجع OSI یک نرم افزار کاربردی نیاز دارد که اطلاعات را در شبکه منتقل کند . اما به خاطر داشته باشیم که عکس این عمل نیز در مقابل و در کامپیوتر مقصد نیز بایستی انجام شود یعنی لایه های OSI در کامپیوتر مقصد عکس همین اعمالی را انجام میدهد که در مبدا انجام می دهند . به فرآیند ایجاد بسته های اطلاعاتی برای ارسال در شبکه Encapsulationیا کپسوله سازی و

به فرآیند از بسته خارج کردن اطلاعات ار بسته های اطلاعاتی Decapsulation یا از کپسول حارج کردن هم می گویند که برخی اوقات ممکن است به عنوان Assemble و Disassembleنیز بکار برده شود .

نتیجه

ویندوز برای ارسال اطلاعات در شبکه از لایه های OSI کمک می گیرد . این خیلی مهم است که درک کنید که لایه های OSI در حقیقت یک مدل کاملا فرضی برای درک فضای شبکه هستند و در حقیقت چنین لایه هایی وجود فیزیکی ندارند ، این مدل برای آموزش و درک بهتر شبکه بهترین مدل مرجع است ، مدل های دیگری مانند TCP IP نیز وجود دارند که دارای چهار لایه می باشند

سنسورها

لودسل

سنسوری الکترونیکی هست که جهت اندازه گیری نیرو و وزن اجسام به کار گرفته میشود

فلو متر یا دبی سنج (flow meter)

وسیله هست که حجم مواد عبوری را نسبت به زمان اندازه گیری می کنه.

در بیشتر صنایع از جمله صنایع نفت و پتروشیمی و... دونستن مقدار دقیق فلو یا جریان عبوری سیالی مانند گاز و یا نفت مهمه. توجه به این نکته که میزان کل فلو که نشون دهنده میزان ماده مصرفی هست تو خیلی جاها مثل پروسه های صنعتی نیاز هست.

اندازه گیری فلو فقط تو کنترل پروسه کاربرد نداره و بعضی جاها برای مسائل مالی استفاده میشه تا فقط مقدار رو بدونن

تکنیک های مختلفی برای اندازه گیری فلو استفاده میشه که میتونم به قسمتی از اصلی ترین هاش اشاره کنیم

+اندازه گیری جریان به روش جابجائی مثبت Positive Displacement

+اندازه گیری جریان به روش اوریفیس پلیت Orifice Plate

+اندازه گیری جریان به روش توربینی Turbine

+اندازه گیری جریان به روتامتری (variable area)

+اندازه گیری جریان به آلتراسونیک با پدیده دوپلر Ultrasonic: Doppler

+اندازه گیری جریان به آلتراسونیک با روش زمان انتقال و .... Ultrasonic: Transit Time, Time of Travel, Time of Flight

+اندازه گیری جریان به ونتوری Orifice Plate

+اندازه گیری جریان به مغناطیسی Magnetic

+اندازه گیری جریان به ورتکس Vortex Shedder

+اندازه گیری جریان به روتامتر) Variable Area (Rotameter

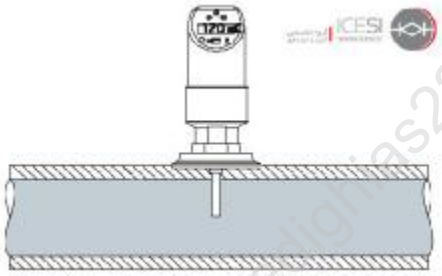
+اندازه گیری جریان به فلومتر مخصوص کانالهای باز Weir and Flume Open channel

+اندازه گیری جریان به کوریالیس Coriolis

+اندازه گیری جریان به گرمائی Thermal
+اندازه گیری جریان به پیتوت Pitot Tube

نمایشگر و فلو سوییچ

فلو یکی از چهار پارامتر مهم در صنایع نفت و گاز می باشد که باید اندازه گیری آن را انجام داد،تجهیزی که قصد معرفی آن را داریم در واقع یک فلو سوییچ ونمایشگر فلو می باشد.
این فلو سوییچ در واقع یک فلومتر حجمی (Mass Flow switch) است که در واقع سوییچ جریان برای نظارت ، نمایش و اندازه گیری نسبی جریان مایعات می باشد.



موارد استفاده :

- ✓ توربین ها
- ✓ سیستم های خنک کاری با آب
- ✓ کمپرسورها
- ✓ مبدل های حرارتی
- ✓ سیستم های روانکاری

مشخصات فنی :

- ✓ اندازه گیری فلو 0.03 تا 3 متر/ثانیه برای مایعات
- ✓ دقت اندازه گیری 1٪
- ✓ دو عدد خروجی دیجیتال
- ✓ یکعد خروجی آنالوگ به صورت سفارشی
- ✓ دارای ip66
- ✓ کارکرد در دمای -40 تا +85 درجه

فلومتر روتامتر یا variable area

اینگونه فلومتر ها بر اساس اندازه گیری جرم سیالات (مایعات و گازها) کار میکنند و استفاده زیادی در صنایع نفت،گاز،پتروشیمی و صنایع مربوطه با آنها دارند.
یکی از مهمترین ویژگی این نوع فلومتر های استفاده آنها در شرایط سخت و دشوار فرایند میباشد.

ویژگی های خاص تجهیز:

1) دارای صـفحه نمایش با اسـکیل مدرج 90 درجه ای
2) کاملا خطی
3) قابلیت کار در دماهای بالا (تا 350 درجه)
4) حداکثر فشار کاری 600 بار
5) قابلیت کنترل جریان
6) توقف جریان برگشتی

البته تعدادی از این مدل ها فقط به صورت نمایشگر بوده اما معمولا آنها دارای خروجی انالوگ جریانی بوده و همچنین از پروتکل های صـنعتی همچون HART - profibus بوده.

لول سوییچ مغناطیسی (Magnetic float level switch)

برای اندازه گیری سـطح و حد بالای سـیالات در مخازن،استخر،حوضچه ها میتوان از فلومترهای مغناطیسی استفاده کرد.

کاربرد : جهت اندازه گیری سطح مایعات

خروجی : خروجی لول سـوییچ ها معمولا به صورت 2 سیمه و یا 3 سیمه میباشد که یک حالت آن NO و حالت دیگر آن NC میباشد.

مناسب برای اندازه گیری مایعات رقیق

لول سوییچ دیاپازونی

جهت اندازه گیری مقدار سطح مخازن و اینکه سطح سیال به چه حدی رسیده میتوان از لول سوییچ ها اسـتفاده کرد،همانطور که از نامش پیدا هست لول سوییچ فقط به شـما یک خروجی دیجیتال میدهد که مشـخص کننده بودن و یا نبودن می باشد.

یعنی در واقع به شـما میگویید که سیال به این سطح رسیده و یا نرسیده

خیلی از نفرات فنی فکر میکنند که فلو سوییچ ها میتوانند به صورت دقیق مقدار سطح سیال را به صورت آنالوگ مشخص کنند که این فکر کاملا اشتباه بوده و فقط ار فلو سوییچ برای مطلع شدن از رسیدن سیال به آن قسمت میباشد.

معمولا این فلو سوییچ ها به صورت 2 و یا 3 سیمه میباشند.

کاربرد لول سوییچ ها :

استفاده در لوله ها

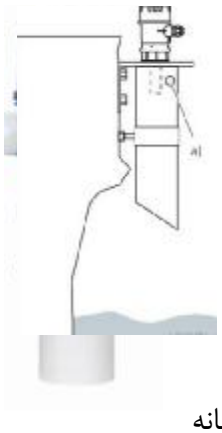
بالا ،وسط و پایین مخازن

لول ترانسمیتر التراسونیک

برای اندازه گیری سطح سیالات به صورت پیوسته در مخازن برای مایعات و مواد فله ای میتوان از ترانسمیتر ها ی التراسونیک استفاده کرد.

سیستم کارکرد :

امواج التراسونیک که به صورت امواجی هستند که در محدوده امواج صوتی بوده که انسان قادر به شنوایی آن نمیباشد ،ترانسمیتر های التراسونیک امواج صوتی را ارسال کرده و منتظر برگشت امواج می میانند ، و با توجه به ثابت بودن سرعت صوت و اندازه گیری زمان رفت و برگشت صوت مقدار فاصله را محاسبه میکنند.



مشخصات فنی تجهیز:

— خروجی آنالوگ به صورت 4 تا 20 میلی آمپر

—اندازه گیری فاصله تا 8 متر برای مایعات

— اندازه گیری فاصله برای مواد فله شده تا 3.5 متر

—دارای ip68

در نصب لول ترانسمیتر باید توجه بسیار به بخارات مایعات داخل مخزن داشته باشید،اگر در دهانه مخازن مایع (مخصوصا مخازن دهانه کوچک)نصب شود ،بخارات باعث ایجاد شبنم بر زیر سنسور میشوند که گاها قطرات و چکه های مایع بر زیر لول ترانسمیتر به وجود آمده ،که این امر دچار اخلال در عملکرد صحیح ترانسمیتر میشود و احتما خطا در اندازه گیری بسیار بالا هست.

سنسور دمای RTD (resistance temperature detector)

می دانیم مقاومت الکتریکی یک جسم با دمای آن رابطه دارد. مقاومت الکتریکی بسیاری از فلزات، با افزایش دما افزایش و با کاهش آن کاهش مییابد. عملکرد RTD ها بر اساس این اصل میباشد. به عبارت دیگر در یک RTD با اندازه گیری مقاومت یک فلز، دمای آن تعیین میشود.

فلز مورد استفاده باید نقطه ذوب بالایی داشته باشد و در مقابل خوردگی مقاوم باشد. پلاتین بهترین فلز برای استفاده در RTD می باشد چون رابطه بین مقاومت و دما کاملاً خطی میباشد.
در مس نیز این رابطه نسبتاً خطی میباشد ولی دامنه اندازه گیری آن کمتر از پلاتین میباشد. در صورت استفاده از نیکل، حساسیت RTD بیشتر خواهد بود ولی برای دماهای بیشتر از ۴۰۰ درجه رابطه به شدت غیرخطی است. از تنگستن میتوان برای اندازه گیری دماهای خیلی بالا استفاده نمود، اما به دلیل مشکلات ساخت و ناپایداری ویژگیهای آن، کمتر مورد استفاده قرار میگیرد.RTD های استاندارد دارای مقاومت پلاتینی هستند که در دمای صفر درجه مقاومت آن ۱۰۰ اهم است و به اصطلاح با آن PT100 گفته میشود. RTD ها دارای انواع دو، سه و چهار سیمه هستند که به ترتیب دقت بالاتری دارند.

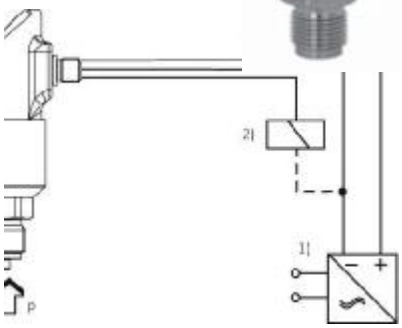
ترانسمیتر دما و رطوبت

جهت اندازه گیری رطوبت و دما در دستگاههای صنعتی،سیستم های اتوماسیون اداری و خانگی و تهویه مطبوع از ترانسمیتر های دما و رطوبت استفاده های بسیاری میشود.

از دیگر مکانهایی که نیاز بسیار زیادی به مانیتورینگ و کنترل دما و رطوبت دارد اتاق های کنترل ،سرور روم ها،مرغداری ها،گلخانه ها،خشک کن های صنعتی و... میتوان نام برد.
این ترانسمیتر ها دارای خروجی های آنالوگ،دیجیتال بوده و پروتکل های صنعتی پشتیبانی میکنند.

پرشر سوییچ دیافراگمی

جهت اندازه گیری و نمایش مقدار فشـار میتوان از تجهیز فوق اسـتفاده کرد البته لازم به ذکر است که تجهیز بالا فقط مقدار فشـار را نمایش میدهد و در خروجی به شـما مقدار فشـار را به صـورت آنالوگ نمیدهد.



این تجهیز درواقع یک پرشـر سـوییچ است که با تنظیم مقدار فشـار میتواند در خروجی آن یک فرمان دیجیتال دریافت کرد،برای مثال شـما میخواهید مقدار فشـار یک مخزن را نمایش دهید و اگر فشار مخزن به 10 بار رسید یک آژیر را به صدا در بیاورید،برای این کار شما میتوانید تجهیز بالا را نصـب کنید و مقدار Setpoint را 10 بار تنظیم کنید به محض اینکه فشـار مخزن بالا رفت در خروجی این تجهیز شـما یک سـیگنال دیجیتال دریافت میکنید که میتوان آن سیگنال را به کار گرفت.

موارد اسـتفاده : اندازه گیری فشـار مایعات،گازها، بخار و گرد و غبار در مخازن یا لوله

انکودر

یکی از تجهیزات کاربردی در اتوماسیون انکدر(Encoder) می باشد.

معنی لغوی آن به معنی رمزگذار است.

بدین صـورت که به وسیله ی ارسال یک سری پالس، سرعت ویامکان شفت موتور را به درایو ویا پی ال سی میدهد، درایو و یا پی ال سی با دی کد کردن یا رمزگشایی پالس ها به سرعت موتور ویا تغییرات مکانی شفت پی خواهد برد.

از این تجهیز می توان به عنوان عنصر فیدبک استفاده نمود.

انکدر در سروو موتورها به صورت یکپارچه در بدنه ی موتور قرار میگیرد.

در دیگر موتورها مانند موتور القایی وموتور DC انکدر به انتهای شفت موتور وصل می شود.

****

روبات های صنعتی

معرفی ربات Yumi ABB

ربات خلاق و انسان دوستانه YuMi که با طراحی بسیار دقیق برای کمک کردن درکارها ی انسان توسط شرکت ABB ساخته شده است و در روز 13آوریل 2015 در هانوفر رسما به دنیا معرفی شد.

این ربات دارای دو بازو می باشد که قابلیت هایی همچون مونتاژ قطعات کوچک و ظریف می باشد که باعث سریعتر شدن پروسه تولید در اسمبل کردن محصولات تولیدی می گردد.

این روبات دارای سیستم ویژن بسیار دقیق، گریپر هایی چابک(پنجه)، فیدبک کنترلی بسیار دقیق، نرم افزاری انعطاف پذیر و ایمن ساخته شده که در مجموع باعث می شود که به جای برنامه نویسی جهت عملکرد ربات فقط کافیست به آن یاد داد که چکار باید انجام دهد.

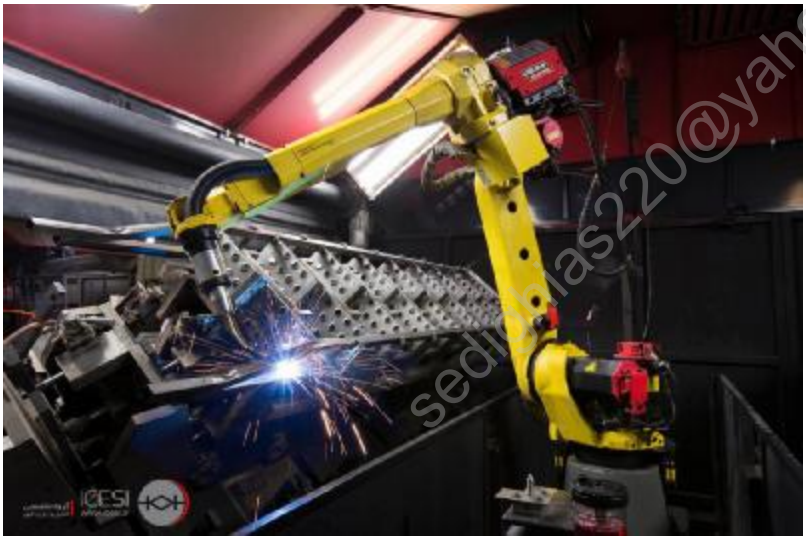
معرفی روبات Flex Picker

این روبات کارایی بسیار زیادی در بهبود بهره وری و انعطاف پذیری عملیات بسته بندی دارد که میتواند تکنیک هوشمندانه ای برای تولید کنندگان باشد. سرعت بالای این روبات استفاده از آن را تبدیل به با صرفه ترین راه حل ممکن در تولید و qc نموده است. این روبات نزدیک به 15 سال است که رهبریت در ربات های بسته بندی را از نظر هنر و سرعت در تکنولوژی بسته بندی بر عهده دارد.

به دلیل سیستم طراحی شده بازوهای آن این امکان را به وجود می آورد تا حرکات سریع و ناگهانی به هر جهت حول مرکزیت روبات را داشته باشد که این قابلیت مختص اینگونه روبات ها می باشد.
یکی از معروفترین مدل های اینگونه روبات ها مربوط به کمپانی ABB با مدل IRB360 بوده تا دارای مشخصات فنی به شرح زیر میباشد:

1) شعاع حرکتی تا 1600 میلیمتر
2) قابلیت اجسام تا 8 کیلو گرم
3) سرعت جابجایی فوق العاده بالا نسبت به بقیه روباتها
4) 3 و یا 4 محوره
5) وزن خود روبات از 120 تا 145 کیلو گرم بوده
6) زمان جابجایی برای 1600 میلی ثانیه با بار 6 کیلو گرم 0.6 ثانیه می باشد.

روبات های جوش



در این تکنولوژی سعی شده نور تولید شده از جوش را به شدت کاسته و ایمنی بصری را در محیط کار بالاتر ببرند. ربات ها در صنایع از نظر هزینه و صرفه اقتصادی در راستای مصرف انرژی و دقت کاری بالا بسیار رواج دارند.

****

معرفی میکروکنترلرها :

به آی سی هایی که قابل برنامه ریزی می باشد و عملکرد آنها از قبل تعیین شده میکروکنترلرگویند میکرو کنترل ها دارای ورودی - خروجی و قدرت پردازش می باشد .

بخشهای مختلف میکروکنترلر :

میکروکنترلر ها از بخشهای زیر تشکیل شده اند

Cpu	واحد پردازش
Alu	واحد محاسبات
I /O	ورودی ها و خروجی ها
Ram	حافظه اصلی میکرو
Rom	حافظه ای که برنامه روی آن ذخیره می گردد
Timer	برای کنترل زمان ها

و . . .

بنابراین میکروکنترلر یک مدار مجتمع یا چیپ الکترونیکی است که دارای CPU,حافظه رم ، رام و تعدادی ورودی خروجی قابل برنامه ریزی است. میکروکنترلر ها در واقع یک میکروکامپیوتر هستند که برای مصارف خاصی برنامه ریزی می شوند. میکرو کنترلر ها در انواع مختلف و برای مصارف مختلفی تولید می شوند.

میکروکنترلر ها توسط کاربر قابل برنامه ریزی هستند که طبق برنامه کاربر می تواند تعریف کند اگر شرایط خاصی در ورودی اتفاق افتاد ، در خروجی اتفاق خاصی بیفتد.

میکرو پروسسور با میکروکنترلر چه تفاوتی دارد؟

احتمالا شما نیز تاکنون از خود پرسیده اید میکروپروسسور با میکروکنترلر چه تفاوتی دارد؟ در پاسخ میتوان گفت میکروپروسسور یک مدار مجتمع پردازشگر است و فاقد هرگونه مدار حافظه و ورودی خروجی جانبی قابل برنامه ریزی است. میکروپروسسور تنها وظیفه پردازش را برعهده دارد و طراح باید خود مدار های حافظه و پورت های ورودی و خروجی را به میکروپروسسور متصل نماید. در حالی که یک میکروکنترلر علاوه بر اینکه شامل یک میکروپروسسور می شود ، در داخل خود شامل حافظه و پورت های ورودی – خروجی قابل برنامه ریزی است.

میکروپروسسور ها اغلب در ساخت رایانه های رومیزی ، لپتاپ و تبلت کاربرد دارند و برای مصارف گسترده ای مورد استفاده کاربران قرار می گیرند. این درحالیست که میکروکنترلر ها اغلب برای ساخت دستگاه ها و سیستم های دارای کاربرد های مشخص به کار گرفته می شوند.

میکروکنترلر AVR چیست ؟

AVR در ابتدا یک خانواده از میکروکنترلر های ۸ بیتی بود که در سـال ۱۹۹۶ برپایه معماری تغییر یافته هاروارد طراحی و ساخته شد و توسط شرکت Atmel روانه بازار های جهانی شد. این میکروکنترلر یکی از پرفروش ترین میکروکنترلر ها در کل جهان به شمار می آید و تاکنون در پروژه های کثیر علمی ، تحقیقاتی و تجاری گوناگونی به کارگرفته شده است.

AVR سـری های مختلفی را شـامل می شـود و فرآیند توسعه این خانواده از میکروکنترلر ها همچنان ادامه دارد. از شناخته شده ترین سـری های AVR می توان به سـری Attiny,Atmega,AtXmega اشاره نمود. در حال حاضر AVR در سری های مختلف و متنوعی با توان های پردازشـی گوناگون ، ظرفیت های حافظه ای مختلف و سرعت پردازش متفاوت در بازار های جهانی موجود است. اما چیزی که در حال حاضر در کشورمان به صورت عمومی و گسترده استفاده می شـود ، سری Atmega از خانواده AVR است که به علت قیمت مناسب و توان پردازش بالا مورد استقبال قرار گرفته است.

نمونه سوالات امتحان

پی ال سی PLC و RTU هر کدام مخفف چه کلماتی در انگلیس هستند و هر کدام چه کارهایی میکند و اختلاف این دو چیست ؟(یک نمره)

مدار منطقی قابل برنامه ریزی Programmable Logic Control یک سیستم فرمان کنترلی برای ماشین‌آلات خودکار میباشد که تجهیزات یک صنعت را توسط سنسورها و مبدل ها و مسیر مخابراتی مشاهده و با برنامه قابل کنترل مینماید

پایانه راه دور Remote Terminal Unit یک سیستم کامپیوتری شامل سخت افزار و نرم افزار جهت جمع‌آوری داده از نقاط مختلف ایستگاه ها (مثلا مقادیر جریان و ولتاژ و وضعیت ها و آلارمها و فرامین) و کد نمودن و برچسب زمانی زدن بر داده ها و ارسال به بالادست (اپراتور ایستگاه - مرکز کنترل دیسپاچینگ) – در حقیقت چنانچه به یک سیستم جمع آوری داده (Data Concentrator) ، قابلیتها و توانمندیهای یک مرکز کنترل کوچک را اضافه نماییم آنرا RTU نامند

داده ها در سیستم اتوماسیون برق چیست(در اتوماسیون برق چه اطلاعاتی رد و بدل میشود) (یک نمره)

چهار داده بین ایستگاه برق و مرکز کنترل دیسپاچینگ بایستی رد و بدل شود شامل : 1- Status ارسال وضعیت تمام کلیدها (باز هستند یا بسته) از ایستگاه برق به مرکز کنترل 2- Alarm ارسال آلارم ها از ایستگاه برق به مرکز کنترل 3- Measurand ارسال مقادیر اندازه گیری ولتاژ و جریان از ایستگاه برق به مرکز کنترل 4- Command ارسال فرمان به کلیدها (بازشوند یا بسته) از مرکز کنترل به ایستگاه

مونیتورینگ چیست و سیستم HMI چیست فرق این دو چیست HMI مخفف چه کلماتی به زبان انگلیسی است(یک نمره)

مونیتورینگ عبارت است از جمع آوری اطلاعات مورد نظر از بخشهای مختلف یک واحد صنعتی و نمایش آنها با فرمت مورد نظر برای رسیدن به اهداف مثل :

- ✓ نمایش وضعیت لحظه ای هر یک از ماشین آلات و دستگاهها
- ✓ نمایش و ثبت پارمترهای مهم و حیاتی یک سیستم
- ✓ نمایش و ثبت آلارمهای مختلف در زمانهای بروز خطا در سیستم

سیستم Human Machin Interface رابط بین ماشین و انسان است، سیستمی است که کاربر(اپراتور) میتواند یک صنعت را مشاهده و یا با فرمان کنترل نماید(شامل نرم افزار و سخت افزار کامپیوتری مخصوص کاربر است)

سیستم بلادرنگ (سر وقت) RTSچیست و کاربرد آن چیست (سه سطر) (دو نمره)

سیستم بلادرنگReal Time System ، یک سیستم کنترل نرم‌افزاری (در خدمت سخت افزار) است در این سیستم نه تنها منطق اجرا، بلکه زمان اجرا بسیار مهم است، به زبان عامیانه سیستم بلادرنگ، سیستمی است که اولا باید درست کار کند و ثانیا آن کار را سر وقت و بدون تاخیر انجام دهد.

مثلا کنترل تجهیزات هواپیما که درون تجهیزات در اتاق خلبان میباشد یا سیستمهای کنترل چراغهای راهنمایی که در دهها چهار راه، در سطح شهرهای بزرگ و یا کنترل و مراقبت از دما در نیروگاهها، سیستم کنترل موتور، سیستمهای پزشکی(مثلا کنترل ضربان قلب)، پردازشگرهای کنترل صنعتی، سیستمهای تلفن همراه، کنترل اسباب بازیها و ...

کاربرد آن در 1- سیستمهای کنترل دیجیتال 2- سیستمهای فرمان و کنترل 3- پردازش سیگنال 4- سیستم ارتباطات راه دور و ...

سیستم کامپیوتر شخصی PC که ما در منزل داریم بلادرنگ نیست

انواع سیستم بلادرنگ (توضیح دوسطر) (دو نمره)

الف)- سیستمهای بلادرنگ سخت : سیستمی که اهمیت زیادی به محدودیت زمانی میدهد و اگر طول زمان از مهلت زمانی خارج شود عمل بی‌فایده است و سیستم ناقص میشود یا یک فاجعه رخ میدهد (مثلا سیستمهایی که با جان انسان ها و با جان تجهیزات سر و کار دارد مثلا سیستم کنترل هواپیما) این سیستمها قابلیت انعطاف و سازگاری کمی دارند اما متداول است که غیر از یک سیستم بلادرنگ سخت که در جلو میباشد یک سیستم غیر بلادرنگ کمکی در پشت موجود باشد. بطورکلی در یک سیستم بلادرنگ سخت حداقل نمودن موارد ذیل

Min (تعداد قیدهای زمانی گمشده) & Min (کل زمان دیرکرد)

ب)– سیستمهای بلادرنگ نرم – سیستمی که محدودیت زمانی دارد اما گاها از دست دادن داده قابل صرفنظر کردن است و منجر به فاجعه نمیشود مثلا سیستمهای پردازش صوت و تصویر که با توجه به محدودیت زمانی خطاهایی درپی دارد که باعث کاهش کیفیت شده اما سیستم متوقف نمیشود و بعمل خود ادامه میدهد در حقیقت در این سیستم قیدهای زمان را حداقل میکنند

هر چند در محاسبات تمرکز بر محاسبات روی سیستمهای سخت میباشد ولی ایجاد سیستم نرم بمراتب پیچیده تر است. بطورکلی در یک بلادرنگ نرم حداقل نمودن موارد ذیل

Min (تعداد قیدهای زمانی) & Min (مقداری از زمان دیرکرد)

دو مدلهای سیستمهای بلادرنگ و سه مورد نیازها در بلادرنگ را بنویسید(دو نمره)

الف) مدل وظیفه بلادرنگ منظم (پریودیک یا دوره ای) - مثلا سیستم در بازه های زمانی منظم درجه حرارت را ثبت کند

ب) مدل وظیفه بلادرنگ نامنظم – در پی یک رخداد مشخص سیستم عکس العمل نشان میدهد مثلا رسیدن دما به یک مقدار مشخص شده یا مثلا پرتاب شدن صندلی هواپیمای جنگنده هنگام حادثه

نیازهای رفتاری: با رفتارهای سیستم، بشکل دوره ای یا برحسب اتفاق وظایفی تعریف میشود

نیازهای زمان موقت: هر چند در سیستم بلادرنگ اساس بر رفتار سیستم است اما بدلیل موقت بودن رفتارها وظیفه ها انحصاری نیست

نیازهای هزینه: تمایل به سود اقتصادی از سیستم ها باعث میشود نرم افزار برای سخت افزار ساخته شود

سیستم عامل در سیستمهای بلادرنگ RTOSچگونه است و دارای چه مشخصاتی است(دو نمره)

در یک سیستم بلادرنگ برای انجام عمل صحیح در سر وقت و بدون تاخیر، مطلوب است که سیستم عامل اصلا موجود نباشد (مثلا از میکرو کنترلر استفاده شود) اگر مجبور به انتخاب سیستم عاملRTOS (Real Time Operating System) شدیم بسیار اهمیت دارد تا هم عمل صحیح و هم سر زمان مشخص، محاسبه و پردازش کند بنابراین ویژگی سیستم عامل اینکه:

1- براحتی قابل برنامه ریزی باشد(انجام هر تعداد عملیات -زمان لختی کم -زمان اجرا کم –نرسیدن به زمان مرگ – اولویت زمانبندی را بشناسد و ..)

2- تحصیص منابع (تخصیص خوب منابع در برنامه ریزی – اولویت بندی)

3- وقفه ها (اینتراپت) را قبول نکند

عملکرد سیستم عامل 1) قطعی بودن – 2) پاسخدهی خوب – 3) قابل کنترل توسط کاربر – 4) قابل اطمینان – 5) نرم در قبال خطا

در طراحی سیستم عامل به دو صورت 1) طراحی براساس اولویت – 2) طراحی بر اشتراک زمان میباشد

در صنایع جهت تنظیم وقت سیگنالهای برای زمان دقیق و همزمان در سیستمها از GPS (General Position System) استفاده میکنند که کلاک (ساعت) یکی از نیازهای سیستم عامل بلادرنگ میباشد. سیستم عامل بلادرنگ، سیستمهای عاملی هستند که همگی آنها از زبانهای سطح بالا استفاده میکنند.

در صنعت از سیستم عاملهای متفاوت استفاده میشود مثلا (IRNX88 - (PORTOS) – (QMX) - (IRNX86) - (Solaris) - (سیستم عامل Unix (Linux ..) (RT Linux=Real Time Linux)) (سیستم عامل ویندوز) (...) (سیستم عامل ویندوز در صنعت قابلیت اعتماد کمی دارد زیرا ویژگیهای فوق را ندارد و آسیب پذیر است) (سیستم عامل لینوکس بدلیل برنامه ریزی در اولویت اجرا و تعاملی بودن و تخصیص منابع متناسب و پایداری، سیستم عامل خوبی است) (بسیار مناسب است که کرنل سیستم عامل قابلیت جاسازی نمودن برنامه کاربردی embedded applicationداشته باشد – مدولار باشد – و ..)

کرنل Kernel– واسط بین برنامه کاربردی و سخت افزارها میباشد که با هر درخواست از برنامه کاربردی، کرنل وظیفه‌ی تخصیص منابع از جمله CPU و Ram و آدرس دهی به حافظه جانبی و .. بعهده دارد

در یک سیستم بلادرنگ دوره ای(پریوریک) از چهار واقعه متناوب با دوره تناوب 100و 100 و 200و500 میلی ثانیه تشکیل شده است اگر هر واقعه بترتیب2و 50و30و100 میلی ثانیه از CPU نیاز داشته باشد آیا سیستم قابل زمانبندی است حداکثر زمان لختی برای کل چهار واقعه چقدر میتواند باشد تا هنوز سیستم قابل زمانبندی باشد(سه نمره)

$$\sum C_i/P_i \leq 1$$

$$\sum \left(\frac{2}{100}\right)+\left(\frac{50}{100}\right)+\left(\frac{30}{200}\right)+\left(\frac{100}{500}\right)=\frac{20}{1000}+\frac{500}{1000}+\frac{150}{1000}+\frac{200}{1000}=\frac{870}{1000}=0.87 \leq 1$$

چون کوچکتر از 1 میباشد پس در یک سیستم قابل زمانبندی است

$$1-0.87=0.13$$

پس کل زمان لختی تمام چهار فعالیت حداکثر 130 میلی ثانیه میتواند باشد

در یک سیستم بلادرنگ پریودیک(دوره ای) از سه واقعه متناوب با فرکانس 10و5و20هرتز تشکیل شده است اگر هر واقعه بترتیب 50و40و15 میلی ثانیه از CPU نیاز داشته باشد با محاسبات نشان دهید آیا سیستم قابل زمانبندی است.($t=1/f$ که f = فرکانس بر حسب هرتز و t = زمان یک دوره بر حسب ثانیه و یک ثانیه = 1000 میلی ثانیه) (سه نمره)

$f_i =$ فرکانس هر واقعه i در سیستم

$C_i = cpu$ زمان پروسس مربوط به واقعه i در

آنگاه شرط پاسخگویی سیستم به کلیه وقایع

$$\sum C_i * f_i \leq 1$$

$$\sum (50/1000*10)+\left(\frac{40}{1000}*5\right)+\left(\frac{15}{1000}*20\right)=\frac{500}{1000}+\frac{200}{1000}+\frac{300}{1000}=\frac{1000}{1000}=1 \leq 1$$

بله چون مساوی یک شد قابل قابل برنامه ریزی است

دریک سیستم بلادرنگ دوره ای سه عملیات با اولویت باید صورت گیرد(سه نمره) عمل با اولویت اول: در فاصله زمانی هر 50 میلی ثانیه بمدت 20 میلی ثانیه(با تاخیر20) (A رنگ آبی یا هاشور خط عمودی) عمل با اولویت دوم: در فاصله زمانی هر 100 میلی ثانیه بمدت 10 میلی ثانیه(با تاخیر 10) (B رنگ قرمز هاشور یا خط افقی) عمل با اولویت سوم: در فاصله زمانی هر 200 میلی ثانیه بمدت 10 میلی ثانیه(با تاخیر30) (C رنگ مشکی یا هاشورمورب) آیا سیستم قابل برنامه ریزی است؟ دیاگرام اولویت بندی و زمانبندی چگونگی عمل ها در CPU را بصورت یک جدول با سطر و ستونهای کامل بنویسد؟

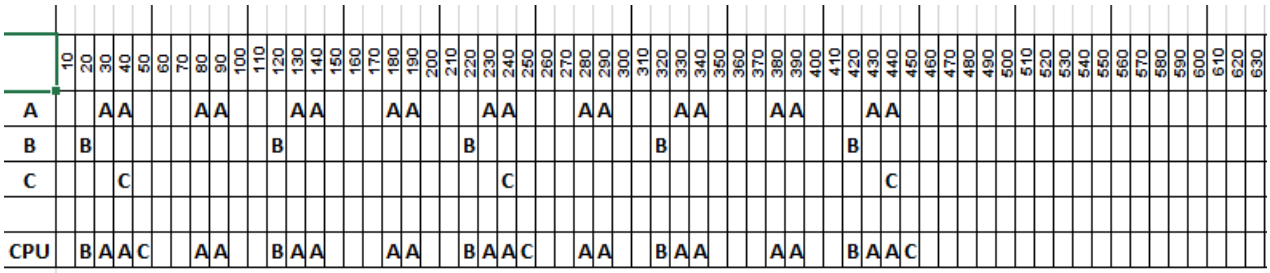
$$\sum C_i/P_i \leq 1$$

$$\Sigma \left(\frac{20}{50}\right)+\left(\frac{10}{100}\right)+\left(\frac{10}{200}\right)=\frac{400}{1000}+\frac{100}{1000}+\frac{50}{1000}=\frac{550}{1000}=0.55$$ پانصد و پنجاه میلی ثانیه

550+20+10+30=610 میلی ثانیه

$0.610 \leq 1$ یا $610 \leq 1000$

چون 610 میلی ثانیه کمتر از هزار میلی ثانیه (= یک ثانیه) شد پس سیستم قابل برنامه ریزی است



	10	20	30	40	50	60	70	80	90	100	110	120	130	140	150	160	170	180	190	200	210	220	230	240	250	260	270	280	290	300	310	320	330	340	350	360	370	380	390	400	410	420	430	440	450	460	470	480	490	500	510	520	530	540	550	560	570	580	590	600	610	620	630
A			A	A				A	A				A	A				A	A				A	A				A	A				A	A				A	A				A	A				A	A				A	A				A	A				A	A				A	A			
B		B										B										B										B										B																															
C				C																				C																				C																															
CPU		B	A	A	C			A	A			B	A	A				A	A			B	A	A	C			A	A			B	A	A				A	A			B	A	A	C																															

اولویت اول - فعالیت A بمدت 20 میلي ثانیه و هر از 50 میلي ثانیه با 20 میلي ثانیه تاخیر
اولویت دوم - فعالیت B بمدت 10 میلي ثانیه و هر از 100 میلي ثانیه با 10 میلي ثانیه تاخیر
اولویت سوم - فعالیتC بمدت 10 میلي ثانیه و هر از 200 میلي ثانیه با 30 میلي ثانیه تاخیر

****

در تهیه این جزوه از تجربیات و اطلاعات شـــخصـــی خودم و نوشـــته ها و نظرات دیگران در اینترنت استفاده نمودم . بدیهی است این جزوه دارای اشکالاتی است که با تذکرات شما و ارسال به آدرس زیر نقایص برطرف و تکمیل میشود. Sedighias220@yahoo.com

****

در هر حرفه ای که هستید نه اجازه دهید که به بدبینیهای بیحاصل آلوده شوید و نه بگذارید که بعضی لحظات تاسف بار که برای هر ملتی پیش می آید شما را به یاس و ناامیدی بکشاند. در آرامش حاکم بر آزمایشگاهها و کتابخانه هایتان زندگی کنید .

نخست از خود بپرسید : " برای یادگیری و خودآموزی چه کرده ام ؟ "

سپس همچنان که پیشتر میروید بپرسید : " من برای کشورم چه کرده ام ؟ "

و این پرســش را آنقدر ادامه دهید تا به این احســاس شــادیبخش و هیجان انگیز برســید که شاید سهم کوچکی در پیشرفت و اعتلای بشریت داشته اید.

اما هر پاداشی که زندگی به تلاشهایمان بدهد یا ندهد هنگامی که به پایان تلاشهایمان نزدیک میشویم هر کداممان باید حق آن را داشــته باشیم که با صدای بلند بگوییم " من آنچه در توان داشته ام انجام داده‌ام "

لوئی پاستور 1895