

هم طراحی سخت افزار نرم افزار

جلسه بیستم: سنتز توأم-زمان بندی ۲

ارائه دهنده: آتنا عبدی

a_abdi@kntu.ac.ir

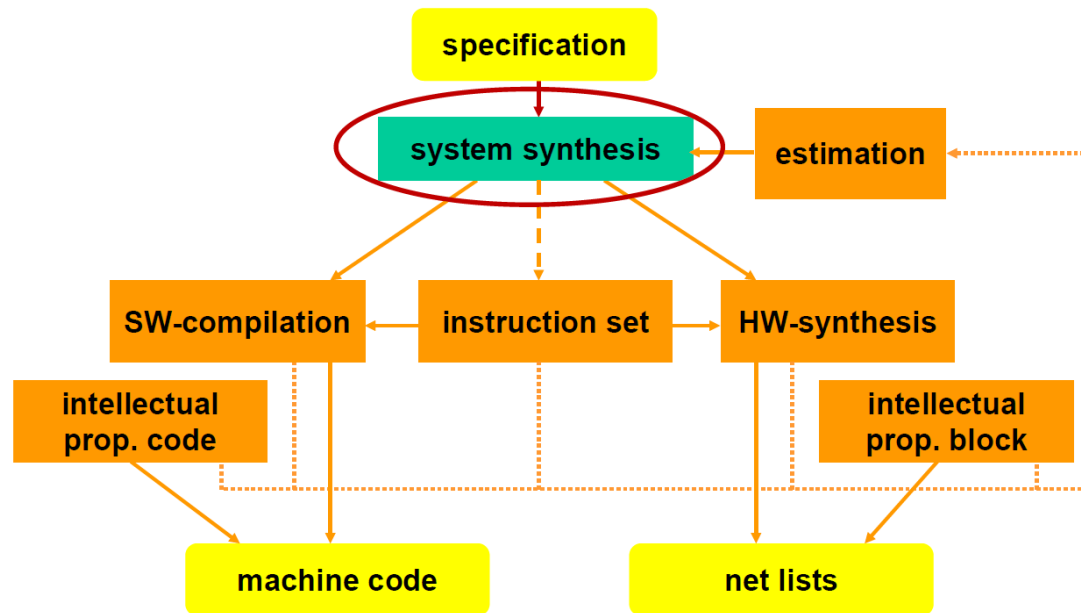
مباحث این جلسه



- سنتز توام در روال هم طراحی سخت افزار و نرم افزار

- زمان بندی (Scheduling)

- رویکردهای زمان بندی با محدودیت



زمان‌بندی با در نظر گرفتن محدودیت منابع



- شکل‌گیری مصالحه بین تاخیر و تعداد منابع در سیستم
- روش‌های قطعی: ILP
- تعریف مسئله در قالب معادله و نامعادله و حل آن
- قابل حل برای حالت‌های محدود و مناسب برای مشخص کردن حدود مسئله
- روش‌های مکاشفه‌ای: مانند زمان‌بندی لیستی (List Scheduling)
- مبتنی بر اولویت‌دهی به گره‌های پردازشی
- در نظر گرفتن وابستگی‌های داده‌های و محدودیت‌های منابع سیستم

الگوریتم زمان‌بندی لیستی (List Scheduling)



- شکل‌گیری الگوریتم براساس ساختمان داده لیست
- محدودیت منابع را در نظر گرفته و براساس گراف وظایف پیش می‌رویم
- وظایف آماده در لیست قرار می‌گیرند
- بکارگیری اولویت در مواردی که محدودیت منابع داریم و می‌بایست انتخاب صورت گیرد
- بهینه‌سازی کارایی با محدودیت منابع یا بهینه‌سازی منابع با محدودیت کارایی امکان‌پذیر است

الگوریتم زمان‌بندی لیستی (ادامه)



- ساخت لیستی از وظایف که آماده اجرا هستند
- براساس وابستگی داده‌ای بین وظایف با در نظر داشتن محدودیت منابع
- مرتب‌سازی وظایف لیست براساس اولویت آن‌ها به صورت نزولی
- حرکت سیکل به سیکل در ترتیب زمان‌بندی
- انتخاب دستورالعمل‌ها براساس محدودیت منابع از ابتدای لیست و زمان‌بندی آن‌ها
- بروزرسانی لیست برای مرحله بعد
- ادامه تا خالی شدن لیست وظایف آماده

الگوریتم زمان بندی لیستی (ادامه)



- تعیین اولویت بین وظایف

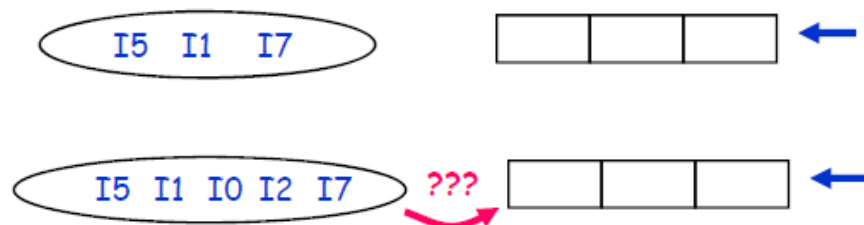
- تعیین وظیفه بحرانی براساس الزامات مسئله

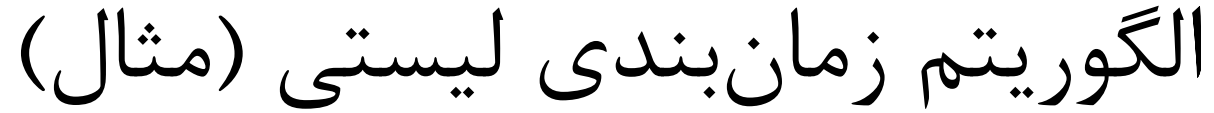
- پرکاربردترین: Mobility

- زمان اجرای در نظر گرفته شده برای هر وظیفه

- موقعیت گره در گراف براساس مسیر بحرانی

- و





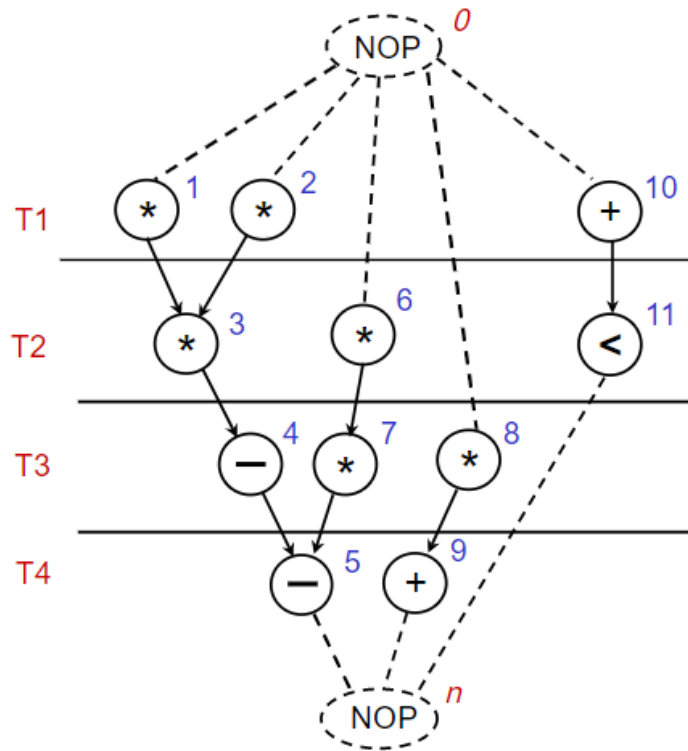
- Ready List = {V1, V2, V6, V8/V10} •

- اولویت‌دهی بر اساس Mobility

- **بروزرسانی لیست وظایف آماده** Ready List = {V3,V6, V8/V11}

- مرحله دوم: انتخاب سه وظیفه مشابه مرحله قبل

- Ready List = {V7,V8, V4} •



الگوریتم زمان‌بندی لیستی (مثال)



- با فرض داشتن دو ضرب‌کننده و دو واحد محاسبات پایه در سیستم

- مرحله سوم:

Ready List = {V7, V8/V4} •

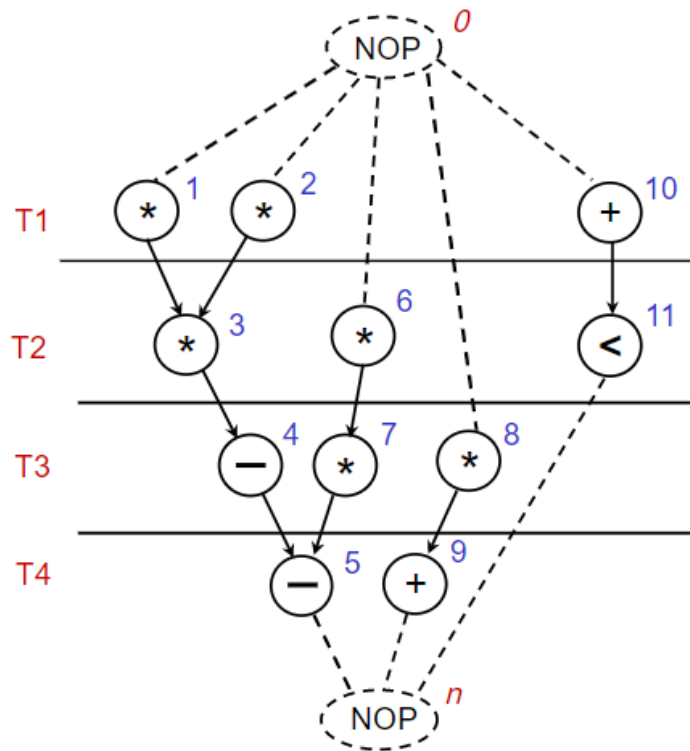
- انتخاب دو وظیفه سر لیست برای ضرب و یکی برای جمع

Ready List = {-/V5, V9} •

- مرحله چهارم:

- زمان‌بندی وظایف و بروز رسانی لیست

Ready List = {} •





الگوریتم زمان‌بندی لیستی (ادامه)

- دیدگاه حریصانه و استاتیک دارد
- حالتی که بررسی کردیم کمینه کردن زمان با داشتن محدودیت منابع بود
- مسئله زمان‌بندی مدنظر ماست
- امکان کمینه کردن منابع با داشتن محدودیت زمان نیز وجود دارد
- مشخص کردن اولویت براساس تفاضل زمان‌بندی ALAP و زمان آماده شدن وظیفه
- تخصیص حداقل منابع براساس اولویت با رعایت محدودیت تاخیر و زمان اجرا

فرایند زمان‌بندی پویا



- اجرای الگوریتم زمان‌بندی در زمان اجرا
- مناسب در سیستم‌ها و کاربردهایی که اطلاعات از سیستم کامل نیست
- سیستم‌های دارای رفتار پویا، متغیر و مبتنی بر تصمیم
- مناسب در کاربردهای نهفته و بی‌درنگ
- اهمیت timeliness و functionality
- مزیت: انعطاف‌پذیری و واقعی بودن

فرایند زمان بندی پویا



- معیارهای حائز اهمیت در الگوریتم زمان بندی پویا
 - قابلیت رعایت درست و به موقع مواعدها
 - پیچیدگی کم و اعمال سربار زمانی محدود به سیستم
 - توزیع متناسب بارکاری بین اجزای پردازشی

الگوریتم‌های زمان‌بندی پویا



- دو روش رایج در حیطه زمان‌بندی پویا و مناسب سیستم‌های بی‌درنگ

- RMS: Rate Monotonic Scheduling

- تعیین اولویت‌ها به صورت ثابت و ایستا

- EDF: Earliest Deadline First

- تعیین اولویت‌ها به صورت پویا

الگوریتم زمان بندی (Rate Monotonic) RM

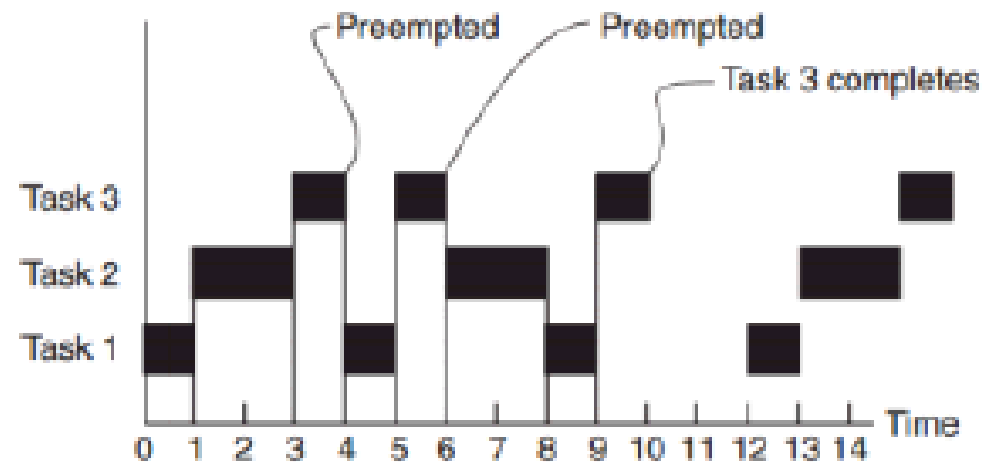


- روش زمان بندی با اولویت ایستا و بهینه که تغییرپذیر نمی باشد
- وظایف از قبل معلوم نیست اما قانون اولویت آن ها مشخص است
- تعریف اولویت براساس دوره اجرای وظایف
- وظیفه با کوتاهترین دوره تناوب (نرخ درخواست)، دارای بیشترین اولویت است
- تخمین دوره اجرا: موعد اجرا
- وظایف در حین کار قابل متوقف شدن می باشند (Preemptive Tasks)
- بهترین کارکرد را نسبت به روش های اولویت ایستا

الگوریتم زمان بندی RM (مثال)



Task	Execution Time	Period	Priority
T1	1	4	High
T2	2	6	Medium
T3	3	12	Low



الگوریتم‌های زمان‌بندی RMS (ادامه)



- بسیار ساده است و فرض می‌کند بین وظایف وابستگی داده‌ای وجود ندارد
- از تمامی توان پردازشی پردازنده استفاده نشده و بعضی جاها خالی می‌ماند
- امکان استفاده از بازه‌های بیکاری پردازنده‌ها برای مدیریت شرایط پیش‌بینی نشده
- سوئیچ کردن زیاد بین وظایف دارد
- فرض می‌شود قابل چشم‌پوشی باشد