

به نام
خدا



درس طراحی و پیاده‌سازی زبان‌های برنامه‌سازی

موضوع: **Sequence Control**

ارائه: صادق سلیمانی

www.Bioinformation.ir

جلسه نهم

عناوین جلسه

- کنترل ترتیب اجرا در عبارات
- کنترل ترتیب اجرا در جملات
- کنترل ترتیب اجرا در زیربرنامه‌ها

عناوین ارائه: کنترل ترتیب اجرا در: عبارات - جملات - زیربرنامه‌ها

سطوح بررسی Sequence Control

1. در داخل Expression ها

- صریح، ضمنی

2. بین Statement ها

- صریح، ضمنی

3. بین زیربرنامه‌ها

1. Call-Return

2. Exception Handlers

3. Scheduler

4. Coroutines

5. Tasks

طراحی و پیاده‌سازی زبان‌های برنامه‌سازی

۳ از ۴۳

عناوین ارائه: کنترل ترتیب اجرا در: عبارات - جملات - زیربرنامه‌ها

ترتیب اجرا در عبارات محاسباتی

• ترتیب اجرا در داخل عبارات

• بر اساس Functional Composition

• استفاده از ساختار درختی

• پرسش:

آیا ترتیب محاسبه برای زیردرخت چپ و راست مهم است؟

• در صورت وجود Side Effect، بلی

• مثال:

• $A+B+Func(C)$

• اگر $Func(C)$ به عنوان مثال، پس از هر بار اجرا، ۲ واحد به A بیفزاید!

• تصمیم‌گیری درباره‌ی پرسش فوق بر عهده‌ی کیست؟

• پیادساز

طراحی و پیاده‌سازی زبان‌های برنامه‌سازی

۴ از ۴۳

عناوین ارائه: کنترل ترتیب اجرا در: عبارات – جملات – زیربرنامه‌ها

بیان‌های خطی ساختار درختی عبارات

1. Prefix Form

1. Ordinary Prefix

- $*(+(A,B),-(A,B))$

2. Cambridge Prefix

- $(* (+ A B) (- A C))$

3. Polish

- $*+AB-AC$

2. Postfix Form

- $AB+AB-*$

3. Infix Form

- $(A+B)*(A-B)$

طراحی و پیاده‌سازی زبان‌های برنامه‌سازی

۴۳ از ۵

عناوین ارائه: کنترل ترتیب اجرا در: عبارات – جملات – زیربرنامه‌ها

بیان‌های خطی عبارات محاسباتی

روش‌های

روش‌های

Postfix & Polish Prefix

Ordinary & Cambridge Prefix

- | | |
|---|---|
| ● عدم نیاز به پرانتز | ● نیاز به پرانتزهای زیاد |
| ● عدم خوانایی از دیدبرنامه‌نویس | ● عدم خوانایی از دیدبرنامه‌نویس |
| ● قابل استفاده با عملگرها با هر تعداد آرگومان | ● قابل استفاده با عملگرها با هر تعداد آرگومان |
| ● سهولت ترجمه توسط کامپایلر | ● سهولت ترجمه توسط کامپایلر |
| ● نیاز کامپایلر به دانستن تعداد عملوندهای هر عملیات | ● عدم نیاز کامپایلر به دانستن تعداد عملوندهای هر عملیات |

طراحی و پیاده‌سازی زبان‌های برنامه‌سازی

۴۳ از ۶

عناوین ارائه: کنترل ترتیب اجرا در: عبارات - جملات - زیربرنامه‌ها

بیان‌های خطی عبارات محاسباتی

روش Infix

- نیاز به پرانتزهای زیاد
- نیاز به قوانین تقدمی رفع ابهام
- در صورت عدم استفاده از پرانتز
- نیاز به Associativity
- در عبارات با عملگرهای همسطح
- خوانایی از دیدبرنامهنویس
- تنها برای عملگرهای دوتایی قابل اعمال است

مثالی از استفاده هر یک در زبان‌های برنامه‌نویسی بزنید؟

طراحی و پیاده‌سازی زبان‌های برنامه‌سازی

۴۳ از ۷

عناوین ارائه: کنترل ترتیب اجرا در: عبارات - جملات - زیربرنامه‌ها

نمایش زمان اجرای عبارات

- Machine Code
- ترجمه مستقیم عبارت به کد ماشین
- امکان تفسیر سخت‌افزاری و اجرای سریع
- Tree Structure
- اجرای مستقیم ساختار درختی توسط مفسر نرم‌افزاری
- مانند زبان لیسپ
- Postfix or Prefix Form
- پویش عبارت از چپ به راست
- محاسبه با کمک پشته

طراحی و پیاده‌سازی زبان‌های برنامه‌سازی

۴۳ از ۸

عناوین ارائه: کنترل ترتیب اجرا در: عبارات - جملات - زیربرنامه‌ها

مسائل مرتبط با محاسبه‌ی نمایش درختی عبارات

1. Uniform Evaluation Rules

- نحوه‌ی تبدیل ساختار درختی به دنباله عبارات مقدماتی یکنواخت باشد
- قوانین ارزیابی یکنواخت باشد
- اما در عمل اینگونه نیست
- Eager Evaluation
 - مثال
 - $(A+B)*(A-C)$
- Lazy Evaluation
 - مثال
 - If $x>0$ then $t=y/x$ else $t=x$

طراحی و پیاده‌سازی زبان‌های برنامه‌سازی

۹ از ۴۳

عناوین ارائه: کنترل ترتیب اجرا در: عبارات - جملات - زیربرنامه‌ها

مسائل مرتبط با محاسبه‌ی نمایش درختی عبارات

2. Side Effects

- مثال
- $A+A*F(n)$
 - مقادیر اولیه: $A=2$ و $F(n)=5$
 - $F(n)$ یک زیربرنامه است که پس از اجرا، یک واحد به A اضافه می‌کند
 - $F(n)$ اثر جانبی دارد
- برخورد زبان‌ها با اثرات جانبی
 1. در برخی زبان‌ها مجاز نیست
 2. در برخی زبان‌ها مجاز است اما یک کامپایلر بسیار دقیق و قوی نیاز دارد

طراحی و پیاده‌سازی زبان‌های برنامه‌سازی

۱۰ از ۴۳

عناوین ارائه: کنترل ترتیب اجرا در: عبارات - جملات - زیربرنامه‌ها

مسائل مرتبط با محاسبه‌ی نمایش درختی عبارات

3. Error Conditions

- $A/B * C$ vs. $A * C / B$
- کدامیک احتمال خطای بیشتری دارد؟
- $A * C / B$ ممکن است با Overflow همراه شود
- تشخیص این قبیل موارد ترتیب اجرا بر عهده‌ی کیست؟
- برنامه‌نویس

4. Short Circuit Boolean Expressions

طراحی و پیاده‌سازی زبان‌های برنامه‌سازی

۱۱ از ۴۳

عناوین ارائه: کنترل ترتیب اجرا در: عبارات - جملات - زیربرنامه‌ها

مسائل مرتبط با محاسبه‌ی نمایش درختی عبارات

4. Short Circuit Boolean Expressions

- If $x > 0$ & $t < 5$ then
- محاسبه‌ی سریعتر چگونه است؟
- اگر شرط اول نادرست باشد نیازی به محاسبه‌ی دومی نیست
- If $x > 0$ or $t < 0$ then
- محاسبه‌ی سریعتر چگونه است؟
- اگر شرط اول درست باشد نیازی به محاسبه‌ی دومی نیست

طراحی و پیاده‌سازی زبان‌های برنامه‌سازی

۱۲ از ۴۳

عناوین ارائه: کنترل ترتیب اجرا در: عبارات - جملات - زیربرنامه‌ها

ترتیب اجرا بین جملات (Statement)

1. بر اساس Composition
 - اجرا به ترتیب فیزیکی
 - اجرا برنامه‌ی بیرونی، اجرای برنامه‌ی درونی را نیز شامل خواهد شد
2. بر اساس Alternation
 - مواردی مانند Case، if...then...else و ...
 - یکی از دو حالت اجرا می‌شود و نه هر دو
3. بر اساس Iteration
 - حلقه‌ها

عناوین ارائه: کنترل ترتیب اجرا در: عبارات - جملات - زیربرنامه‌ها

ترتیب اجرا بین جملات (Statement)

4. مبتنی بر Goto
 1. شرطی
 - If Cond. Then Goto L1
 2. غیرشرطی
 - Goto L1
- محاسن GoTo
- معایب GoTo

عناوین ارائه: کنترل ترتیب اجرا در: عبارات - جملات - زیربرنامه‌ها

برنامه‌سازی ساخت‌یافته Structured Programming

● تعریف

- یک روش برنامه‌سازی است که تأکید دارد بر
 - طراحی برنامه به صورت سلسله‌مراتبی
 - بیان این سلسله‌مراتب با دستورات کنترلی ساخت‌یافته مانند while, for و ...
 - پرهیز از نوشتن روتین‌های چندمنظوره
 - اجرای هر دستور در موقعیت فیزیکی خودش

طراحی و پیاده‌سازی زبان‌های برنامه‌سازی

۱۵ از ۴۳

عناوین ارائه: کنترل ترتیب اجرا در: عبارات - جملات - زیربرنامه‌ها

کنترل ترتیب اجرا در برنامه‌سازی ساخت‌یافته

● Structured Control Statements

- دارای خصوصیت one-in one-out هستند
- کنترل را از دستور قبلی خود دریافت نموده و به دستور بعدی خود منتقل می‌کنند

1. Compound Statement

- فرم کلی

Begin

-Sequence of statements one or more

End

- پیاده‌سازی:
- به سادگی و با حذف Begin و End و اجرای دستورات متوالی، توسط کامپایلر

طراحی و پیاده‌سازی زبان‌های برنامه‌سازی

۱۶ از ۴۳

عناوین ارائه: کنترل ترتیب اجرا در: عبارات - جملات - زیربرنامه‌ها

کنترل ترتیب اجرا در برنامه‌سازی ساخت‌یافته

2. Conditional Statement

3. Case Statement

4. Iteration Statement

1. حلقه ساده

PERFORM body K TIMES

مسائل:

۱. اگر مقدار K منفی یا صفر باشد، چه تصمیمی گرفته شود؟
 - حلقه اصلاً انجام نشود
 - حلقه یک بار انجام شود
 - خطا و توقف برنامه
۲. اگر K در حین اجرای body تغییر کرد، چه تأثیری بر تعداد تکرارها داشته باشد؟
 - معمولاً K یک بار محاسبه می‌شود (تغییرات تأثیری نداشته باشد)

طراحی و پیاده‌سازی زبان‌های برنامه‌سازی

۱۷ از ۴۳

عناوین ارائه: کنترل ترتیب اجرا در: عبارات - جملات - زیربرنامه‌ها

کنترل ترتیب اجرا در برنامه‌سازی ساخت‌یافته

4. Iteration Statement

2. حلقه با برقراری یک شرط

- **While <شرط> Do <عبارت>**

- تغییر متغیر مرتبط با شرط حلقه بر عهده‌ی برنامه‌نویس است

3. حلقه با افزایش یک شمارنده

- **For** مقدار نهایی; افزایشده; مقدار اولیه=متغیرکنترلی

مسائل

- در صورت درست نبودن مقادیر کنترلی چه باید کرد؟
- در صورت تغییر مقادیر کنترلی در حین اجرا، چه باید کرد؟

طراحی و پیاده‌سازی زبان‌های برنامه‌سازی

۱۸ از ۴۳

عناوین ارائه: کنترل ترتیب اجرا در: عبارات - جملات - زیربرنامه‌ها

کنترل ترتیب اجرا در برنامه‌سازی ساخت‌یافته

Iteration Statement 4.

2. حلقه بی‌پایان

loop

Exit when شرط

End loop

طراحی و پیاده‌سازی زبان‌های برنامه‌سازی

۱۹ از ۴۳

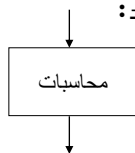
عناوین ارائه: کنترل ترتیب اجرا در: عبارات - جملات - زیربرنامه‌ها

برنامه‌نویسی بدون Goto

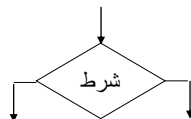
• فلوجارت

• گرافی که سه نوع گره ممکن است داشته باشد:

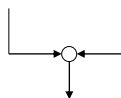
1. Functional node



2. Decision Node



3. Join Node



طراحی و پیاده‌سازی زبان‌های برنامه‌سازی

۲۰ از ۴۳

عناوین ارائه: کنترل ترتیب اجرا در: عبارات - جملات - زیربرنامه‌ها

برنامه‌سازی ساخت‌یافته

● Proper Program

- فلوچارتی است که یک پیکان ورودی و یک پیکان خروجی دارد،
- گره‌های آن از سه‌نوع ذکر شده هستند
- از پیکان ورودی به همگی ندها یک مسیر داریم
- از همگی ندها به پیکان خروجی نیز مسیر داریم

● Prime Program

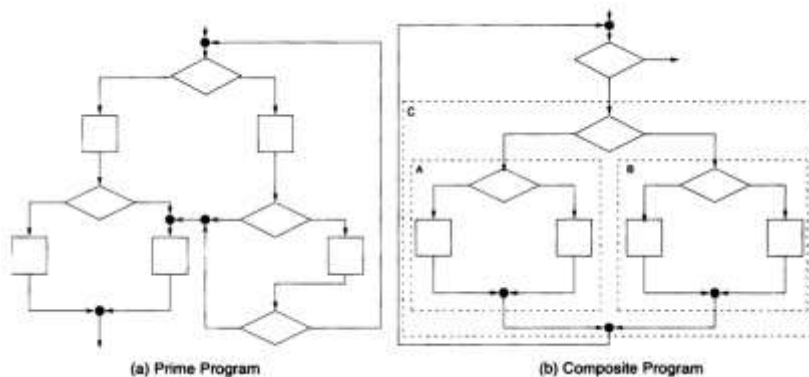
- یک Proper Program است که نتوان آن را به Proper Program های دیگر تجزیه کرد

طراحی و پیاده‌سازی زبان‌های برنامه‌سازی

۲۱ از ۴۳

عناوین ارائه: کنترل ترتیب اجرا در: عبارات - جملات - زیربرنامه‌ها

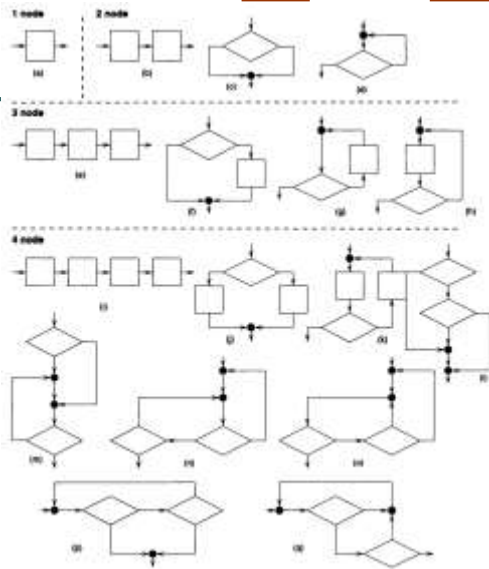
مفهوم Prime Program



طراحی و پیاده‌سازی زبان‌های برنامه‌سازی

۲۲ از ۴۳

عناوین ارائه: کنترل ترتیب اجرا در: عبارات - جملات - زیربرنامه‌ها



● Prime Program های اولیه

طراحی و پیاده‌سازی زبان‌های برنامه‌سازی

۲۳ از ۴۳

عناوین ارائه: کنترل ترتیب اجرا در: عبارات - جملات - زیربرنامه‌ها

برنامه‌سازی ساخت‌یافته

- مواردی که ناگزیر به استفاده از Goto هستیم
 1. Multiple Exit Loops
 2. Loop and a Half
 3. Exception Conditions

طراحی و پیاده‌سازی زبان‌های برنامه‌سازی

۲۴ از ۴۳

عناوین ارائه: کنترل ترتیب اجرا در: عبارات - جملات - زیربرنامه‌ها

کنترل اجرا بین زیربرنامه‌ها

1. (Copy-Rule) Call and Return
2. Recursive
3. Exception Handler
4. Coroutines
5. Schedulers
6. Tasks

طراحی و پیاده‌سازی زبان‌های برنامه‌سازی

۲۵ از ۴۳

عناوین ارائه: کنترل ترتیب اجرا در: عبارات - جملات - زیربرنامه‌ها

زیربرنامه‌های Call and Return

1. پیش شرط‌ها
 - زیربرنامه بازگشتی نباشد
 - در غیر این صورت زیربرنامه از نوع بازگشتی است
 - زیربرنامه صریحا فراخوانی شود
 - در غیر این صورت Exception Handler داریم
 - زیربرنامه پس از فراخوانی از ابتدا تا انتها اجرا شود
 - در غیر این صورت Coroutine داریم
 - زیربرنامه پس از فراخوانی بلافاصله اجرا شود
 - در غیر این صورت Scheduler داریم
 - در هر لحظه تنها یک رشته در ستورالعمل در حال اجرا داشته باشیم
 - در غیر این صورت در حال استفاده از Task ها هستیم

طراحی و پیاده‌سازی زبان‌های برنامه‌سازی

۲۶ از ۴۳

عناوین ارائه: کنترل ترتیب اجرا در: عبارات - جملات - زیربرنامه‌ها

زیربرنامه‌های Call and Return

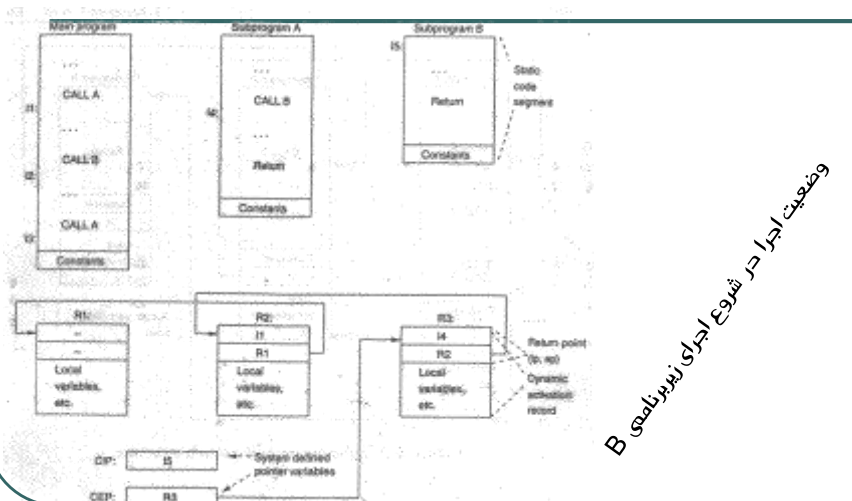
- پیاده‌سازی مبتنی بر دو رجیستر
 - CIP (Current Instruction Pointer)
 - آدرس دستورالعمل جاری
 - CEP (Current Environment Pointer)
 - آدرس Activation Record جاری

طراحی و پیاده‌سازی زبان‌های برنامه‌سازی

۲۷ از ۴۳

عناوین ارائه: کنترل ترتیب اجرا در: عبارات - جملات - زیربرنامه‌ها

زیربرنامه‌های Call and Return



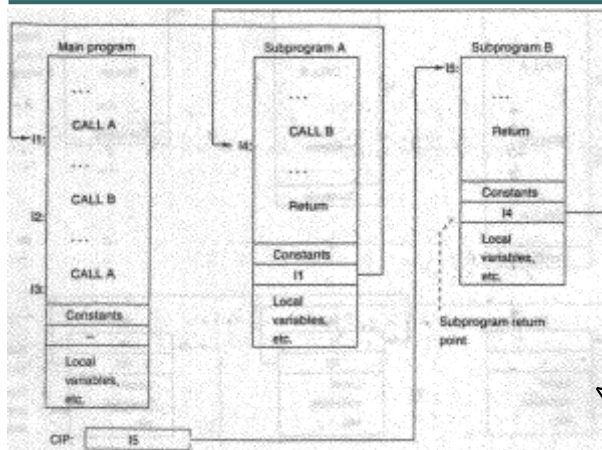
وضعیت اجرا در شروع اجرای زیربرنامه B

طراحی و پیاده‌سازی زبان‌های برنامه‌سازی

۲۸ از ۴۳

عناوین ارائه: کنترل ترتیب اجرا در: عبارات - جملات - زیربرنامه‌ها

زیربرنامه‌های Call and Return



اختصاص ایستای حافظه
Activation Record

طراحی و پیاده‌سازی زبان‌های برنامه‌سازی

۴۳ از ۲۹

عناوین ارائه: کنترل ترتیب اجرا در: عبارات - جملات - زیربرنامه‌ها

زیربرنامه‌های Call and Return

- مزیت اختصاص ایستای حافظه به Activation Record
 - افزایش سرعت
 - نیاز به CEP نداریم
- عیب اختصاص ایستای حافظه به Activation Record
 - هدر رفتن حافظه

طراحی و پیاده‌سازی زبان‌های برنامه‌سازی

۴۳ از ۳۰

عناوین ارائه: کنترل ترتیب اجرا در: عبارات - جملات - زیربرنامه‌ها

زیربرنامه‌های بازگشتی

- فراخوانی خود یا دیگری
- پیاده‌سازی مشابه به زیربرنامه‌های Call and Return است اما یک تفاوت
- ممکن است از Activation Record یک برنامه چند کپی داشته باشیم
- معمولاً از Runtime Stack در مدیریت حافظه استفاده می‌شود

طراحی و پیاده‌سازی زبان‌های برنامه‌سازی

۴۳ از ۳۱

عناوین ارائه: کنترل ترتیب اجرا در: عبارات - جملات - زیربرنامه‌ها

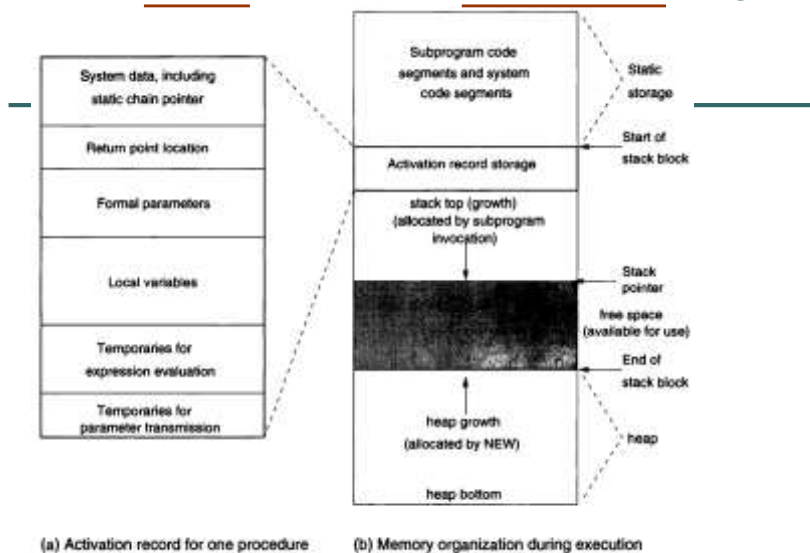


Figure 9.3. C memory organization.

طراحی و پیاده‌سازی زبان‌های برنامه‌سازی

۴۳ از ۳۲

عناوین ارائه: کنترل ترتیب اجرا در: عبارات - جملات - زیربرنامه‌ها

Exception Handlers

- شرایط استثنایی
- نیاز به فراخوانی زیربرنامهای برای انجام عملیات ویژه مرتبط با استثنا
- فراخوانی به صورت ضمنی است (نیاز به نام یا پارامتر ندارد)
- رسیدگی به مواردی مانند زیر:
 - وضعیت خطا (Error Condition)
 - مانند سرریز محاسباتی
 - یا ارجاع به اندیسی خارج از محدوده‌ی آرایه
 - Tracing and Monitoring
 - در خلال تست برنامه، مانند چاپ خروجی برنامه در حین تست
- شرایط پیش‌بینی نشده (Unpredictable)
- وقوع در اجرای عادی برنامه؛ مانند رویداد EOF در فایل ورودی

طراحی و پیاده‌سازی زبان‌های برنامه‌سازی

۴۳ از ۲۳

عناوین ارائه: کنترل ترتیب اجرا در: عبارات - جملات - زیربرنامه‌ها

Exception Handlers

- استثناها به وسیله سخت‌افزار، سیستم عامل یا کامپایلر کشف و رسیدگی می‌شوند
- گاهی خود کامپایلر، روتین‌هایی برای مدیریت آن‌ها فراهم نموده است
- گاهی به برنامه نویسی اجازتی تهیمی روتین مربوطه داده می‌شود
- مثال
 - زبان Ada

```

procedure Sub is
  Bad_Data_Value: exception;
  --other declarations for Sub
begin
  --statements for normal processing in Sub
exception
  when Bad_Data_Value =>
    --handler for bad data values
  when Constraint_Error =>
    --handler for predefined exception Constraint_Error
  when others =>
    --handler for all other exceptions
end;

if X = 0 then raise Bad_Data_Value end if;
  
```

طراحی و پیاده‌سازی زبان‌های برنامه‌سازی

۴۳ از ۲۴

عناوین ارائه: کنترل ترتیب اجرا در: عبارات - جملات - زیربرنامه‌ها

Exception Handlers

• مثال

• زبان C++

```
try {
    statement1;
    statement2;
    ...
    if BadCondition { throw ExceptionName};
}
catch ExceptionName { // Do something for exception
    } // End of Exception
}
```

طراحی و پیاده‌سازی زبان‌های برنامه‌سازی

۴۳ از ۳۵

عناوین ارائه: کنترل ترتیب اجرا در: عبارات - جملات - زیربرنامه‌ها

Exception Handlers

• برخورد با استثنا

• در زبان C++

• پس از رسیدگی به Exception، اجرای برنامه قطع می‌شود

• در زبان Ada

• پس از رسیدگی به Exception، برنامه‌ای که در آن Exception روی

داده، قطع شده و به برنامه‌ی فراخواننده‌ی آن برگشت می‌شود

• در زبان PL/I

• پس از رسیدگی به Exception، اجرای برنامه بعد از موقعیتی که

Exception اتفاق افتاده، از سرگرفته می‌شود

طراحی و پیاده‌سازی زبان‌های برنامه‌سازی

۴۳ از ۳۶

عناوین ارائه: کنترل ترتیب اجرا در: عبارات - جملات - زیربرنامه‌ها

Exception Handlers

• انتشار استثنا

- استثنا ممکن است در نقطه‌ی وقوع رسیدگی نشود!
- اگر ترتیب فراخوانی برنامه‌ها به شکل زیر باشد

A -> B -> C -> D -> E

- اگر استثنا در برنامه‌ی E اتفاق بیفتد اما در C رسیدگی شود
- گوییم: استثنا از برنامه‌ی E به C انتشار یافته است
- اگر استثنا در هیچ یک از برنامه‌های فراخوانی شده یافت نشد، Exception Handler خود زبان فعال می‌شود
- برای یافتن یک Exception Handler، Dynamic Change of Program Calls پی‌گیری می‌شود؛ چگونه؟
- با کمک CIP و CEP

طراحی و پیاده‌سازی زبان‌های برنامه‌سازی

۴۳ از ۳۷

عناوین ارائه: کنترل ترتیب اجرا در: عبارات - جملات - زیربرنامه‌ها

Coroutines

- لزومی نیست که زیربرنامه پس از فراخوانی از ابتدا تا انتها اجرا شود!
- برای ایجاد برخی الگوهای همزمانی اجرا مانند نخ‌ها
- کنترل اجرا صریحا با نام بردن به دیگری منتقل می‌شود
- برای برخی شبیه‌سازی‌های گسسته نیز استفاده می‌شوند
- کمتر به صورت مستقیم در اختیار کاربر قرار می‌گیرند بلکه در اختیار زبان برنامه‌سازی هستند

طراحی و پیاده‌سازی زبان‌های برنامه‌سازی

۴۳ از ۳۸

عناوین ارائه: کنترل ترتیب اجرا در: عبارات - جملات - زیربرنامه‌ها

Schedulerها

- بلافاصله پس از فراخوانی اجرا نمی‌شوند
- طبق یک برنامه‌ی زمانبندی اجرا می‌گردند
- فرمت‌های فراخوانی زیربرنامه‌های زمانبندی شده
 - اجرا قبل یا بعد از سایر زیربرنامه‌ها
 - اجرا با شرط عبارت منطقی
 - اجرا بر مبنای یک مقیاس زمانی
 - اجرا بر اساس اولویت
- مثال
 - شبیه‌سازی سیستم‌های گسسته
 - SIMULA ،SIMSCRIPT ،GPSS

طراحی و پیاده‌سازی زبان‌های برنامه‌سازی

۳۹ از ۴۳

عناوین ارائه: کنترل ترتیب اجرا در: عبارات - جملات - زیربرنامه‌ها

Taskها

- اجرای غیرترتیبی
- برنامه‌نویسی موازی
 - در هر لحظه بیش از یک رشته دستورات در حال اجرا داریم
 - افزایش پیچیدگی طراحی زبان
 - مانند: دسترسی چندین فرایند به یک داده، به صورت همزمان
- حالات اجرا
 1. سیستم کامپیوتری توزیع شده
 2. چندپردازنده‌ای
 3. تک پردازنده‌ای

طراحی و پیاده‌سازی زبان‌های برنامه‌سازی

۴۰ از ۴۳

عناوین ارائه: کنترل ترتیب اجرا در: عبارات - جملات - زیربرنامه‌ها

Taskها

• اصول زبان‌های برنامه‌سازی موازی

1. تعریف متغیرها
2. ترکیب موازی
3. ساختار برنامه
4. ارتباطات
5. همزمانی

طراحی و پیاده‌سازی زبان‌های برنامه‌سازی

۴۱ از ۴۳

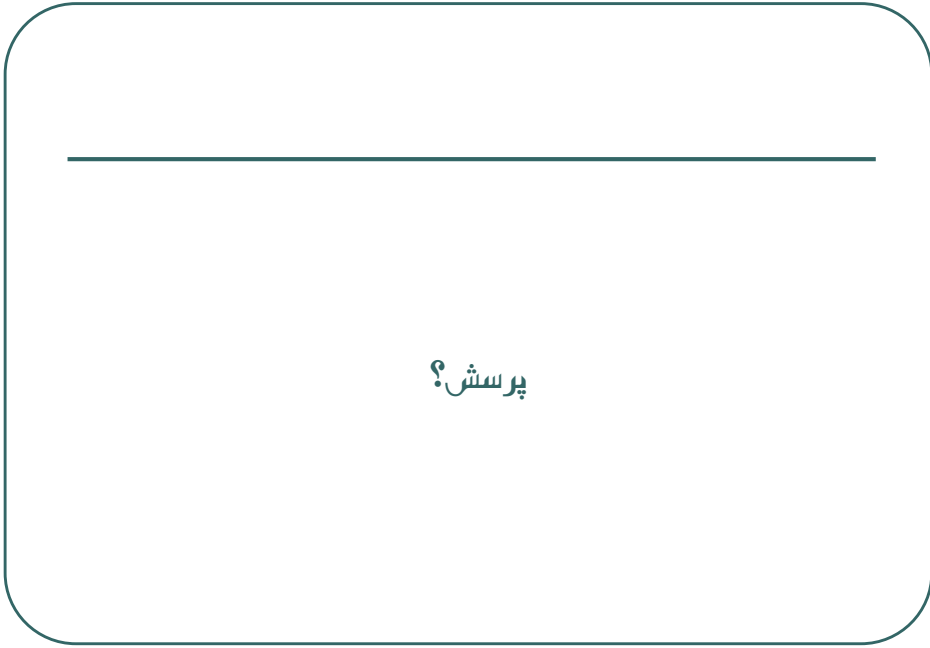
عناوین ارائه: کنترل ترتیب اجرا در: عبارات - جملات - زیربرنامه‌ها

همزمانی بین Taskها

1. استفاده از وقفه
2. سمافور
3. Guarded Statement

طراحی و پیاده‌سازی زبان‌های برنامه‌سازی

۴۲ از ۴۳



طراحی و پیاده‌سازی زبان‌های برنامه‌سازی

۴۳ از ۴۳