دانشگاه صنعتی امیرکبیر (پلی تکنیک تهران)
دانشکدهٔ مهندسی کامپیوتر و فنآوری اطلاعات

اصول و مفاهیم سیستم عامل

ترم مهر ۹۵

فصل چهارم: بن بست

نستوه طاهری جوان

nastooh@aut.ac.ir

تعریف بن بست

- وضعیتی که در آن فرآیندها، هر یک منتظر دیگری باشند، بن بست نام دارد.
  - به عبارت دیگر یک مجموعه از فرآیندها در صورتی دچار بن بست می شوند که هر یک از فرآیندهای این مجموعه **منتظر** رویدادی هستند که فقط فرآیند دیگری از همین مجموعه می تواند آن را ایجاد کند.
  - به عنوان مثال فرض کنید، فرآیند P0 منبع R0 را در اختیار دارد و برای اتمام کار خود به منبع R1 نیز نیاز دارد. از طرفی فرآیند P1 منبع R1 را در اختیار دارد و برای اتمام کار خود به منبع R0 نیز نیاز دارد. در این حالت بن بست رخ داده است.

nastooh@aut.ac.ir

شرایط وقوع بن بست

- ✓ بن بست فقط وقتی می تواند رخ دهد که چهار شرط کافمن به طور همزمان برقرار باشند: ***(یکی از شروط نقض شود، بن بست رخ نمی دهد)***
  - انحصار متقابل (Mutual Exclusion)
    - در آن واحد فقط یک فرآیند می تواند از منبع استفاده کند. (استفادهٔ اشتراکی از منبع نداریم!!)
  - گرفتن و انتظار (Hold & Wait)
    - فرآیندی که یک منبع را در اختیار دارد، می تواند برای منبع جدیدی نیز درخواست دهد و در این مدت منبع قدیمی را نمی تواند رها کند.
  - غیر قابل پس گرفتن (non Preemption)
    - فرآیندها منابع را داوطلبانه رها می کنند. (نمی توان منبع را به زور پس گرفت)
  - انتظار چرخشی (Circular Wait)
    - زنجیره ای از فرآیندها موجودند که هر فرآیند منتظر منبعی است که در اختیار فرآیند دیگری از همان زنجیره است.

nastooh@aut.ac.ir

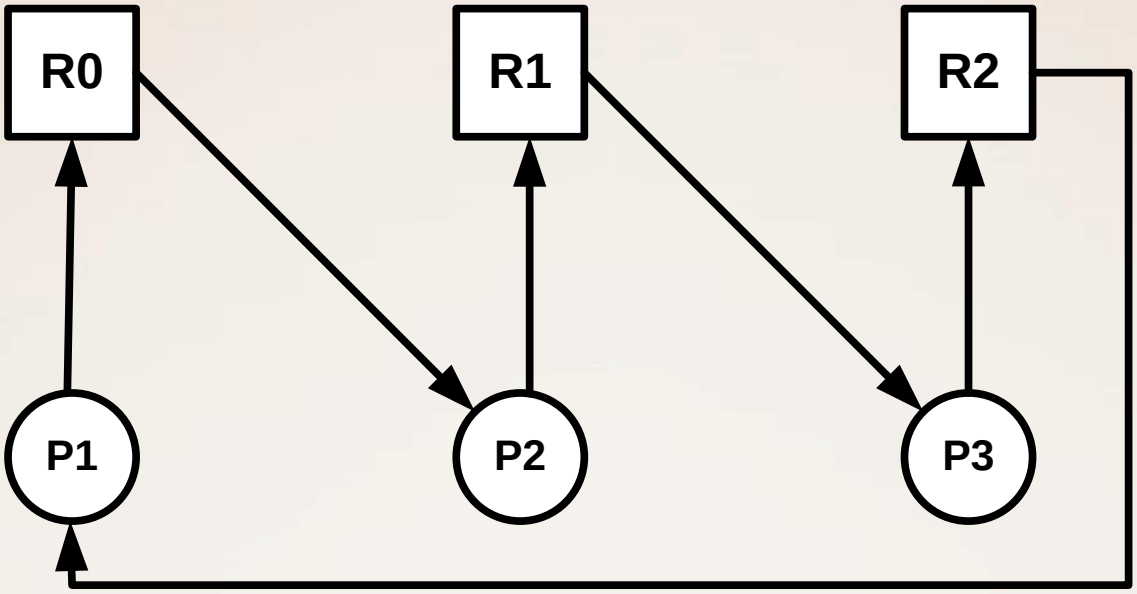
گراف تخصیص منابع

- ✓ برای بررسی شرط انتظار چرخشی از یک گراف به صورت زیر استفاده می کنیم
  - این گراف مجموعه ای از لبه ها و راس ها است.
  - رئوس این گراف دو نوع هستند:
    - فرآیندها، که با دایره نمایش داده می شوند.
    - منابع، که با مربع نمایش داده می شوند.
  - لبه های این گراف به صورت جهت دار و دو نوع هستند:
    - از یک منبع به یک فرآیند، یعنی منبع به فرآیند اختصاص یافته است.
    - از یک فرآیند به یک منبع، یعنی فرآیند به آن منبع نیاز دارد. (درخواست)
  - **اگر از یک منبع چند نمونه در سیستم موجود است، در داخل راس مربوط به آن منبع از چند نقطهٔ پر رنگ استفاده می کنیم.**

nastooh@aut.ac.ir

گراف تخصیص منابع

- اگر گراف تخصیص منابع فاقد سیکل باشد، هیچ فرآیندی در بن بست نیست. (در صورتی که از هر منبع فقط یک نمونه وجود داشته باشد)

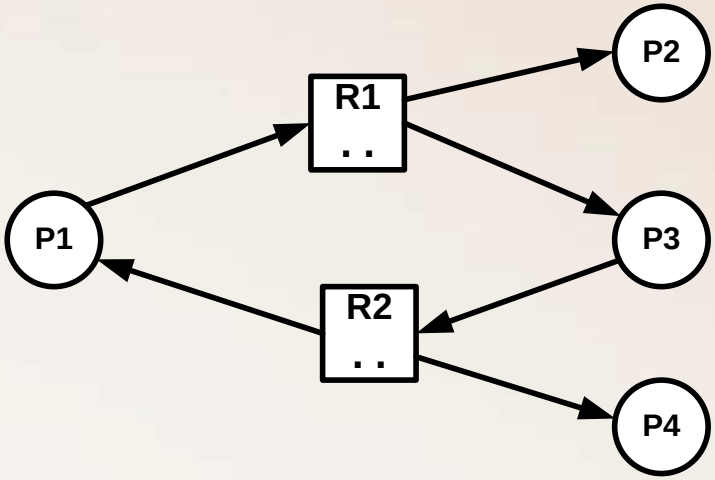


شرط انتظار چرخشی در گراف تخصیص منابع

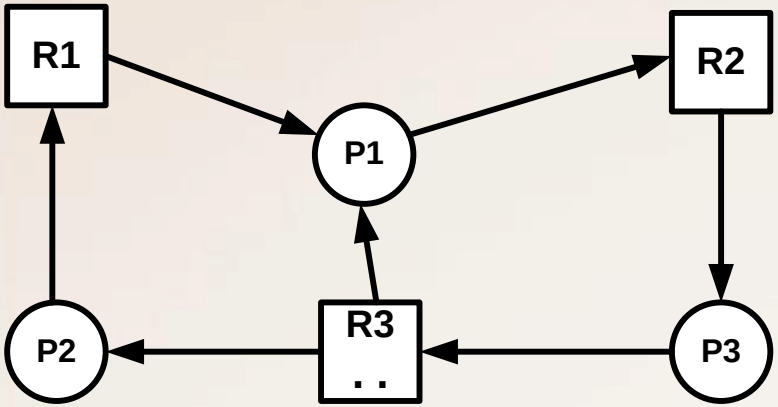
nastooh@aut.ac.ir

گراف تخصیص منابع

✓ در کدام گراف شرط انتظار چرخشی وجود دارد؟



عدم وجود شرط انتظار چرخشی



وجود شرط انتظار چرخشی

چرا؟؟

nastooh@aut.ac.ir

روش های مقابله با بن بست

- ✓ صرف نظر از بن بست!
  - الگوریتم شترمرغ (ostrich)
- ✓ کشف و ترمیم بن بست (Detection and Recovery)
- ✓ پیشگیری از بن بست (Deadlock Prevention)
  - از بین بردن یکی از شروط کافمن
- ✓ اجتناب از بن بست (Deadlock Avoidance)
  - الگوریتم بانکداران

nastooh@aut.ac.ir

الگوریتم شترمرغ

- ✓ نادیده گرفتن بن بست
  - به جهت هزینهٔ بالای روش های مقابله با بن بست، هیچ راهکاری اندیشیده نمی شود.
  - در صورت وقوع بن بست، سیستم دوباره از ابتدا شروع به کار می کند.
  - در سیستم هایی که بن بست به ندرت رخ می دهد شاید بتوان استفاده کرد.

کشف و ترمیم بن بست

- ✓ هیچ هزینه ای برای جلوگیری از بن بست پرداخت نمی شود.
  - اجازه داده می شود تا بن بست رخ دهد، پس از کشف بن بست، باید سیستم از بن بست خارج شود.
- ✓ سوال های اساسی:
  - با چه الگوریتمی و چگونه متوجه بن بست شویم؟
  - الگوریتم تشخیص بن بست در چه زمانهایی اجرا شود؟
  - چگونه از بن بست خارج شویم؟

nastooh@aut.ac.ir

کشف و ترمیم بن بست

- ✓ تشخیص بن بست
  - حالت اول، از هر منبع یک نمونه وجود دارد:
    - با استفاده از گراف تخصیص منابع و چک کردن وجود حلقه در آن
  - حالت دوم، نمونه های متعددی از یک منبع وجود دارد:
    - همانند الگوریتم بانکدار (رجوع شود به مبحث اجتناب از بن بست)
- ✓ اجرای الگوریتم تشخیص بن بست
  - متاثر از پارامترهای زیادی است. به عنوان مثال می توان:
    - هنگامی که یک منبع درخواستی قابل اعطا نیست، اجرا شود.
    - هنگامی که بهره وری پردازنده از یک آستانه کمتر شد، اجرا شود.
    - در بازه های زمانی مشخص اجرا شود.
    - در زمان های تصادفی یا حتی دلخواه اجرا شود.
    - هنگامی که بار پردازشی سیستم پایین است، می توان الگوریتم را اجرا کرد.

کشف و ترمیم بن بست

- ترمیم بن بست
  - اتمام فرآیند
    - خاتمهٔ یک فرآیند و آزاد شدن همهٔ منابع در اختیار آن، و اجرای دوبارهٔ فرآیند
      - هزینهٔ بسیار بالا
      - تاثیر نحوهٔ انتخاب فرآیندها بر روی کارآیی سیستم. (مهم)
  - پس گرفتن منابع
    - منابع را به زور از فرآیندها پس گرفت و فرآیند به عقب برگردانده شود.
      - باز هم تاثیر انتخاب فرآیندها بر روی کارآیی سیستم.

nastooh@aut.ac.ir

پیشگیری از بن بست

- ✓ نقض شرط اول کافمن (انحصار متقابل)
  - ○ استفادهٔ اشتراکی از منابع!
    - برخی از منابع ماهیتاً نمی توانند اشتراکی استفاده شوند. مانند پرینتر.
    - استفاده از تکنیک هایی مانند Spooling برای برخی از منابع.
    - برای برخی دیگر از منابع راه حل عملی نمی توان پیدا کرد.
- ✓ نقض شرط دوم کافمن (نگهداری و انتظار)
  - ○ فرآیندها در ابتدا تمام منابع خود را درخواست دهند و فقط وقتی شروع به کار کنند که همهٔ منابع آزاد باشد.
    - ایراد یک: احتمال گرسنگی
    - ایراد دو: استفادهٔ غیر بهینه از منابع (کاهش بهره وری منابع)
    - ایراد سه: پیش بینی نیازهای آتی گاهی غیر ممکن است.

nastooh@aut.ac.ir

پیشگیری از بن بست

- ✓ نقض شرط سوم کافمن (غیر قابل پس گرفتن)
  - ○ سیستم عامل بتواند در صورت نیاز منابع را به زور از فرآیندها پس بگیرد.
    - عیب: سربار زیاد، زیرا فرآیندها مجبور به دوباره کاری می شوند.
    - در عمل شاید نتوان همهٔ منابع را به زور از فرآیندها پس گرفت.
- ✓ نقض شرط چهارم کافمن (انتظار چرخشی)
  - ○ ایجاد محدودیت در تقاضای فرآیندها.
    - به عنوان مثال منابع را شماره بندی کنیم. سپس فرآیندها فقط بتوانند منابع را به صورت صعودی در اختیار بگیرند. در واقع اگر فرآیندی منبع شماره ۵ را در اختیار دارد، دیگر نمی تواند منبع ۴ را درخواست دهد!
      - ➢ آیا فرآیند می تواند منابع خود را پیش بینی کند؟

nastooh@aut.ac.ir

اجتناب از بن بست

- ایدهٔ اصلی
  - هنگامی که یک فرآیند یک منبع را درخواست کرد، حتی اگر آن منبع آزاد است، فوراً آن منبع را به فرآیند ندهیم، بلکه ابتدا بررسی کنیم با دادن این منبع، در آینده احتمال وقوع بن بست هست، یا خیر؟
- سیستم باید همواره در حالت امن (Safe State) باشد.
  - یک سیستم در حالت امن است اگر یک دنبالهٔ امن وجود داشته باشد.
- دنبالهٔ امن:
  - دنباله ای از فرآیندها به صورت <P1, P2, …, Pn> را دنبالهٔ امن گویند اگر برای هر یک از Pi ها، منابعی که Pi نیاز دارد به وسیلهٔ منابع آزاد فعلی به علاوهٔ منابعی که Pj ها (j<i) در اختیار دارند، تامین شود.
    - مثلا اگر منابعی که P4 نیاز دارد، بلافاصله در اختیار نباشند، P4 می تواند تا زمانی که P1 و P2 و P3 پایان می یابند (و منابع خود را آزاد می کنند) صبر کند.

nastooh@aut.ac.ir

اجتناب از بن بست

- ✓ اگر در یک سیستم هیچ دنبالهٔ امنی وجود نداشته باشد، سیستم در حالت نا امن است.
- ✓ اگر سیستم در حالت امن باشد، هیچ گاه بن بست رخ نمی دهد اما اگر در حالت نا امن باشد **ممکن** است (نه لزوماً) بن بست رخ دهد.

nastooh@aut.ac.ir

اجتناب از بن بست

✓ مثال

○ در یک سیستم سه فرآیند P0 و P1 و P2 و جمعاً ۱۱ عدد نوارخوان وجود دارند. وضعیت فرآیندها و منابع در لحظهٔ جاری (t0) به صورت زیر است:

	حداکثر نیاز	منابع در اختیار	نیاز آینده
P0	۹	۴	۵
P1	۵	۲	۳
P2	۹	۲	۷

- در t0 فقط ۳ منبع آزاد است.
- در لحظهٔ t0 سیستم در حالت امن است. با توالی امن <p1,p0,p2>
- در لحظهٔ t1 فرآیند P2، درخواست ۱ نوارخوان دارد. اگر با درخواست وی موافق شود، با اینکه هنوز ۲ منبع دیگر آزاد هستند، اما سیستم به حالت نا امن می رود.
  - در این حالت نباید منبع مورد درخواست P2 به آن داده شود. و P2 بهتر است منتظر بماند تا طبق دنبالهٔ امن <p1,P0,P2> پیش برویم!

nastooh@aut.ac.ir

الگوریتم بانکدار(ان)

- ✓ فرض: سیستم دارای n فرآیند و m نوع منبع است
- ✓ ساختمان داده های مورد نیاز:
  - آرایهٔ یک بعدی با طول m با عنوان Available
    - تعداد منابع آزاد برای هر منبع را نشان می دهد.
  - ماتریس دو بعدی با اندازهٔ m*n با عنوان MAX
    - حداکثر نیاز هر فرآیند به هر منبع را نشان می دهد.
  - ماتریس دو بعدی با اندازهٔ m*n با عنوان Allocation
    - مشخص می کند در حال حاضر از هر منبع چه تعدادی به به هر فرآیند اختصاص یافته است.
  - ماتریس دو بعدی با اندازهٔ m*n با عنوان Need
    - نشان می دهد هر فرآیند در آینده ممکن است چه تعداد از هر منبع را درخواست دهد.

nastooh@aut.ac.ir

الگوریتم بانکدار(ان)

✓ تشخیص امن بودن یا نا امن بودن وضعیت سیستم

1. جستجو در ماتریس Need به دنبال سطری که همهٔ عناصر آن از Available کوچکتر باشد.
   - در واقع دنبال فرآیندی می گردیم که با منابع آزاد سیستم بتوانیم تمام نیازهایش را مرتفع کنیم.
   - اگرچنین سطری یافت نشود، سیستم در وضعیت نا امن قرار دارد!
2. اگر در مرحلهٔ ۱ سطری پیدا شد، اعداد مربوط به آن فرآیند در ماتریس Allocation را به بردار Available اضافه می کنیم.
   - این فرآیند را به عنوان یک فرآیند خاتمه یافته به حساب می آوریم.
3. مراحل ۱ و ۲ را با بردار به روز شدهٔ Available تکرار می کنیم تا زمانی که همهٔ فرآیندها خاتمه یابند.
   - در این حالت سیستم در وضعیت امن قرار دارد.
   - ترتیب خاتمهٔ فرآیندها به عنوان یک دنبالهٔ امن محسوب می شود.

nastooh@aut.ac.ir

الگوریتم بانکدار(ان)

- اگر فرآیندی درخواست در اختیار گرفتن منابعی را داد، ابتدا به طور ضمنی فرض می کنیم منابع به آن فرآیند اختصاص یابد. حال با مقادیر جدید ماتریس ها، به دنبال یک دنبالهٔ امن می گردیم. اگر کماکان سیستم در وضعیت امن قرار داشت، منابع مورد درخواست فرآیند را به فرآیند اختصاص می دهیم و مقادیر ماتریس ها را به طور قطعی تغییر می دهیم.
- گاهی اوقات در عمل نمی توان از الگوریتم بانکدار استفاده کرد، زیرا
  - فرآیندها معمولاً از نیاز نهایی خود به منابع آگاهی ندارند.
  - تعداد فرآیندها معمولا تغییر می کند.
- الگوریتم بانکدار دارای مرتبهٔ اجرایی $O(m*n^2)$ است.

nastooh@aut.ac.ir

الگوریتم بانکدار(ان)

- مثال: سیستمی پنج فرآیند P0 تا P4 و سه منبع A و B و C دارد. منبع A دارای ۱۰ نمونه، منبع B دارای ۵ نمونه و منبع C دارای ۷ نمونه است. فرض کنید در زمان t0 وضعیت سیستم مطابق جدول های زیر باشد.

Allocation	A	B	C
P0	0	1	0
P1	2	0	0
P2	3	0	2
P3	2	1	1
P4	0	0	2

MAX	A	B	C
P0	7	5	3
P1	3	2	2
P2	9	0	2
P3	2	2	2
P4	4	3	3

Available		
A	B	C
3	3	2

nastooh@aut.ac.ir

الگوریتم بانکدار(ان)

✓ ادامهٔ مثال:

○ آیا سیستم در وضعیت امن قرار دارد؟

- ماتریس Need را محاسبه می کنیم:

Need	A	B	C
P0	7	4	3
P1	1	2	2
P2	6	0	0
P3	0	1	1
P4	4	3	1

- پس از اجرای مراحل الگوریتم بانکدار، دنبالهٔ امن زیر تشکیل می شود، پس سیستم در وضعیت امن قرار دارد:

<P1,P3,P4,P2,P0>

nastooh@aut.ac.ir

الگوریتم بانکدار(ان)

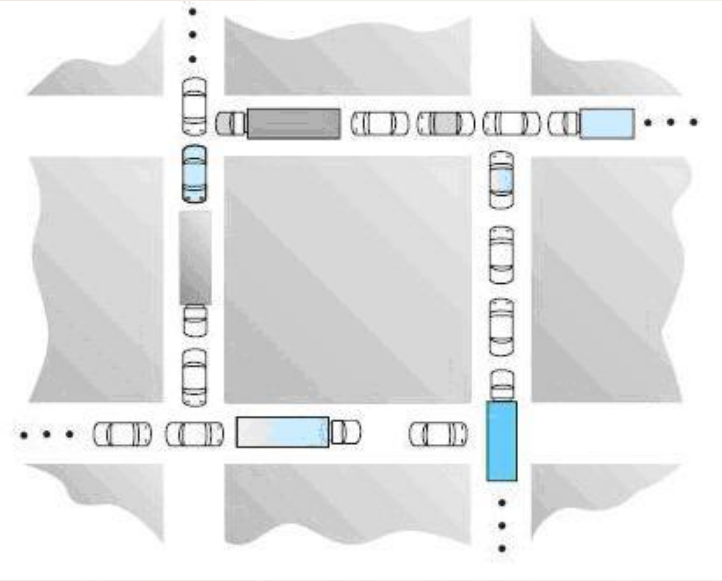
✓ ادامهٔ مثال:

- در لحظهٔ t0 فرآیند P1 یک نمونه دیگر از منبع A و دو نمونه از منبع C را درخواست می کند. آیا با درخواست P1 موافقت کنیم؟
  - بله، زیرا پس از این درخواست دنبالهٔ امن <P1,P3,P4,P0,P2> برقرار است.
- در لحظهٔ t0 فرآیند P4 چهار نمونه دیگر از منبع A و سه نمونه از منبع Bرا درخواست می کند. آیا با درخواست P4 موافقت کنیم؟
  - خیر. چنین منابعی آزاد نیستند.
- در لحظهٔ t0 فرآیند P0 سه نمونه دیگر از منبع B را درخواست می کند. آیا با درخواست P0 موافقت کنیم؟
  - خیر، با اینکه منابع موجود هستند، اما سیستم به وضعیت نا امن می رود.
- در لحظهٔ t0 فرآیند P3 یک نمونه دیگر از منبع A و یک نمونه از منبع Bرا درخواست می کند. آیا با درخواست P3موافقت کنیم؟
  - خیر، منابع درخواستی از ماتریس Need بزرگتر هستند. در واقع چنین درخواستی تعریف نشده است.

nastooh@aut.ac.ir

تمرین ها

- ✓ تمرین ۱: در مورد شکل زیر به سوالها پاسخ دهید.
  - آیا بن بست رخ داده است؟
  - چهار شرط کافمن را به طور دقیق در این مورد بررسی کنید.
  - برای اجتناب از بن بست، قانون (یا قانون هایی) وضع کنید.
    - هنگام ارائهٔ راهکار به پارامترهایی مانند سادگی راه حل، قابل پیاده سازی بودن و بهره وری آن فکر کنید.



nastooh@aut.ac.ir

تمرین ها

- ✓ تمرین ۲: فرض کنید برای مقابله با بن بست، شش راه کار زیر وجود دارد. سوال های الف و ب را کاملا تحلیل کنید.
  - الگوریتم بانکدار
  - کشف بن بست، سپس از بین بردن کامل فرآیندهای دخیل
  - رزرو تمام منابع قبل از شروع به اجرا
  - شروع مجدد فرآیند (و آزاد کردن منابع) در صورتی که فرآیند منتظر یک منبع است.
  - دادن شماره ترتیبی به منابع و گرفتن صعودی یا نزولی منابع توسط فرآیندها
  - کشف بن بست و به عقب برگرداندن فرآیندها.
- ○ الف) این راه حل ها را بر اساس پارامترهای زیر رتبه بندی کنید.
  1. کارآیی پردازنده (هر چه سربار پردازنده کمتر باشد، بهتر است)
  2. اجرای موازی فرآیندهای بیشتر (هر چه فرآیندهای بیشتری در حال اجرا باشند، بهتر است)
- ○ ب) آیا نرخ وقوع بن بست در نحوهٔ رتبه بندی شما تاثیری دارد؟

nastooh@aut.ac.ir

تمرین ها

- ✓ تمرین ۳: راه کار زیر را برای مسالهٔ فیلسوف های خورنده، هم از نظر بن بست و هم از نظر گرسنگی کاملا تحلیل کنید.
  - فیلسوف گرسنه ابتدا چنگال سمت راست را بر میدارد. اگر چنگال سمت چپش آزاد بود، آن را نیز برداشته و شروع به خوردن می کند، در غیر این صورت چنگال سمت راست را نیز روی میز می گذارد!

- ✓ تمرین ۴: به نظر شما آیا یک سیستم می تواند تشخیص دهد برخی از فرآیندها دچار **گرسنگی** شده اند؟ اگر پاسخ شما مثبت است، راهکار خود را تشریح کنید. اگر پاسخ منفی است، سیستم چگونه باید با مسالهٔ گرسنگی کنار بیاید؟

nastooh@aut.ac.ir

تمرین ها

- تمرین ۵: به نظر شما ایدهٔ اصلیِ پشت الگوریتم بانکدار در زندگی روزمره هم کاربرد دارد؟ دربارهٔ سودمندی (یا عدم سودمندیِ) این ایده در دنیای واقعی کمی بحث کنید و مثال بزنید.

nastooh@aut.ac.ir

نمونه تست های کنکور ارشد

✓ ارشد، آزاد، ۸۲

- سیستمی شامل ۵ فرآیند P1 تا P5 و منابع R1 و R2 در حالت زیر مفروض است. حداقل مقدار x برای اینکه سیستم امن باشد کدام است؟

Allocation	R1	R2
P1	0	2
P2	2	5
P3	4	0
P4	1	1
P5	0	0

MAX	R1	R2
P1	5	2
P2	2	10
P3	4	5
P4	1	4
P5	9	5

Available	
R1	R2
x	3

الف) ۳ ب) ۴ ج) ۵ د) ۶

nastooh@aut.ac.ir

منابع

[1]. A. Silberschatz, P. B. Galvin and G. Gagne, **"Operating System Concepts,"** 9th ed., John Wiley Inc., 2013.

[2] A. S. Tanenbaum and H. Bos, "**Modern Operating Systems,**" 4rd ed., Pearson, 2014.

[3] W. Stallings, **"Operating Systems,"** 8th ed., Pearson, 2014.

[4] A. S. Tanenbaum, A. S. Woodhull**, "Operating Systems Design and Implementation,"** 3rd ed., Pearson, 2006.

[5]نستوه طاهری جوان و محسن طورانی، **"اصول و مفاهیم سیستم عامل،"** انتشارات موسسه آموزش عالی پارسه، ۱۳۸۶.

nastooh@aut.ac.ir

پایان

nastooh@aut.ac.ir