

process	Allocation $r_1, r_2, r_3, r_4$	Max $r_1, r_2, r_3, r_4$	Need $r_1, r_2, r_3, r_4$	available $r_1, r_2, r_3, r_4$ <del>2, 1, 1, 2</del>
P1	0, 0, 1, 2	0, 0, 1, 2	0, 0, 0, 0	2, 1, 0, 0
P2	2, 0, 0, 0	2, 2, 0, 0	0, 2, 0, 0	2, 1, 1, 2
P3	0, 0, 2, 2	4, 0, 0, 4	4, 0, 2, 2	
P4	2, 2, 0, 2	2, 2, 0, 4	0, 0, 0, 2	
P5	0, 2, 2, 2	0, 4, 0, 2	0, 2, 2, 0	

4) ادل ۱۰۰۰ (۱۰۰۰۰۰۰۰) نیاز دارد وقتی ابل ۱۰۰۰ (۱۰۰۰۰۰۰۰) رسیده باشد که  $available = available + (۱۰۰۰۰۰۰۰)$  را آزاد کند ←

جواب ۲۰، ۲۱، ۲۲، ۲۳، ۲۴، ۲۵

(2) برای احضار  $\mu$  باید منابع در دسترس (بود و ۷ و ۵) باشند اما

پس آدمی کفعم ای پیرسین را .

③ یہی اجنبی ہم باہر ضائع کافی راکھ ہائیم وی (۲، ۱، ۱، ۲) (۲، ۱، ۲، ۱) اور → اورشی کی کینم میچم سرکاری

④ برای اجرای P4 باید (۲، ۱، ۲) < (۲، ۰، ۰) باشد که درست است ← می توانیم

اجرای کنیم

موضوع:

تاریخ: / /

وقتی P4 اجرای آن allocate و available = available +

$$\rightarrow \text{available} = (2, 1, 2) + P4 \text{ های } (4, 3, 0) = (6, 4, 2)$$

باید منابع دفعه قبل P2 و P3 نمی توانند اجرا شوند

⑤ صریح سراغ P5 ← منابعی که allocate کرده بود، آزاد می کند

$$\text{available} = (5, 3, 3, 2) + (4, 4, 6, 6) = (9, 7, 9, 8)$$

⑥ بابت منابع از راه می توانیم P2 را اجرا کنیم چون

$$(9, 7, 9, 8) < (4, 7, 0, 0)$$

پس از اجرای P2 و منابعی که گرفته بود آزاد می آید

$$\text{available} = (9, 7, 9, 8) + (2, 0, 0, 0) =$$

$$(11, 7, 9, 8)$$

⑦ بابت میزان منابع برای هر پروسه P2 را اجرا کرد چون

$$(11, 7, 9, 8) < (4, 6, 2, 2)$$

بعد از اجرای available آید می شود

$$\text{available} = (11, 7, 9, 8) + (4, 3, 0, 0) =$$

$$(15, 10, 9, 8)$$

چون این وضعیت safe است ← بنی رست

کل منابع در دسترس سیستم در آن کار داریم

دنباله‌ی (P3, P2, P4, P1) و ترتیب اجرای پروسه‌ها

پروسه‌ها

تفاوت در صورتی احتمال ایجاد بن بست وجود دارد که وضعیت safe نباشد

چون با منابع اولیه‌ی (۲، ۰، ۰، ۰) و جدول resource در بن بست هیچ پروسه‌ای نمی تواند اجرا شود

اجرا کردند ← سیستم در وضعیت safe قرار دارد بن بست نداریم

②

③



② یعنی (۳۰۰، ۱۰۰) + (۴، ۳۰۰، ۵۰) = allocation p۴

موضوع: (۴، ۳۰۰، ۱۰۰) تاریخ: / /

تست ۵ p۴ برای رسوایی های (۵۰۰، ۱۰۰) درخواست می دهد

این مقدار را با available می کنیم چون

$\Rightarrow (۵۰۰، ۱۰۰) < (۲۰۰، ۱۰۰)$

① available را آپدیت می کنیم و با بایده می کنیم (مثلاً ای از پرسش ها هست که

safe جان با خیر ۸ (۳۰۰، ۱۰۰) - (۵۰۰، ۱۰۰) = available

② میزان درخواستی را با allocation p۴ = (۲۰۰، ۱۰۰)

افزود می کنیم available need Allocation

	r1	r2	r3	r4	r1	r2	r3	r4	r1	r2	r3	r4
p1	۵	۵	۱	۲	۵	۵	۵	۵	۲	۵	۵	۵
p2	۲	۵	۵	۵	۵	۷	۷	۵				
p3	۵	۱	۳	۴	۶	۶	۲	۲				
p4	۲	۲	۷	۴	۲	۵	۵	۲				
p5	۵	۳	۴	۲	۵	۳	۲	۵				

⑦ p6 برای ته چون  $(۲۰۰، ۱۰۰) < (۵۰۰، ۱۰۰)$

available = (۲۰۰، ۱۰۰) + (۴، ۳۰۰، ۵۰) = (۲۰۴، ۴۰۰، ۱۵۰)

② p1 را می توانیم اجرا کنیم چون (۵۰۰، ۱۰۰) < (۲۰۴، ۴۰۰، ۱۵۰)

③ می ریم برای p۲ ← p۳ هم می تونه اجرا کرد چون

$(۴، ۳۰۰، ۵۰) < (۶، ۶، ۲، ۲)$

④ p4 را می کنیم (۲۰۴، ۴۰۰، ۱۵۰) < (۲۰۴، ۴۰۰، ۱۵۰) ← اجرا می کنیم

و available را آپدیت می کنیم

available = (۲۰۴، ۴۰۰، ۱۵۰) + (۲، ۳، ۷، ۴) = (۴، ۳، ۶، ۶)

⑤ p5 را هم می کنیم چون need < available این اجرای نوز

پس از اجرای ۵ available = (۴، ۳، ۶، ۶) + (۲، ۳، ۷، ۴) = (۸، ۶، ۹، ۸)

(۵، ۷، ۶، ۸) (۴، ۶، ۹، ۸)

ارامی ⑤  $P_2$  را می‌کنیم چون  $need < available$

این ← غیبه ایداری کرد

می‌رویم بدایع  $P_2$  و می‌کنیم چون

(۴، ۶، ۹، ۸)  $<$  (۶، ۶، ۲، ۱)

$available$

$need$

این هم ایداری نور

پس (نبایه ای) نداریم که شامل بهی بدایه ها باشد چون بدایه های

$P_2$  و  $P_3$  نتوانستند ایداری شوند ~~بدایه ها~~ وضعیت

safe نیست پس اگر درخواست  $P_3$  جواب (دفعه ۵) وارد وضعیت

unsafe  $unsafe$  می‌شود (وضعیت) بن بست دارد (چون است و کی قطعاً نیست)

بدایه های  $P_2$  و  $P_3$  چنان است که ایداری بن بست شوند

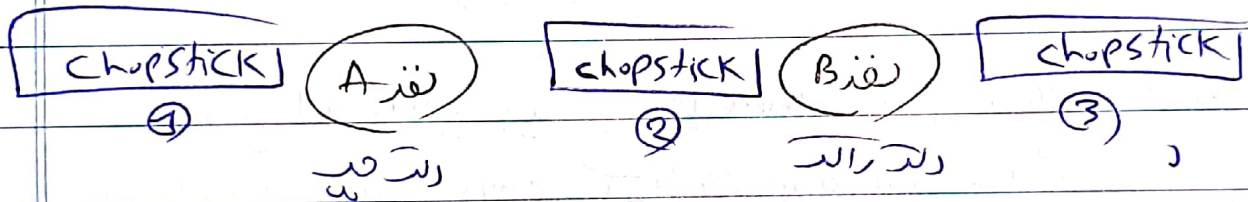
احتمال وجود بن بست  $\rightarrow P_2, P_3$

② چون بهی بدایه ها ایداری و وضعیت safe نور ← احتمال

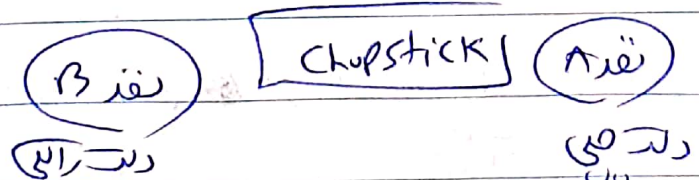
ایجاد بن بست نداریم



سوال ۲ الف) جلوگیری از این بست ← زمانی در راک فواید دلت که  
 wait cycle در آنجا باید و همی فیلسوف هادر cycle باشند و مستقر ریسری  
 باشند که دست به نذر دلت و نذر دلت هم مستقر ریسری باشند که دست نذر دلت  
 و همی chopstick ها باید ~~باشند~~ باشند و hold نباشند  
 وقتی که سر من هم فیلسوف راک دلت دارم هم چید دلت و فقط و دلت  
 یک خفت (رأس) د (ایا فیلسوف) هادر هستند که یکی راک دلت و یکی چید دلت  
 است ← چون همی chopstick ها باید hold شوند ← یکی از این دو نذر  
 باید chopstick بین صورتی (دو یکی) را بردارد



حالت اول: اگر نذر A زودتر chopstick 2 را بردارد ← چون چید دلت است  
 فقط قبلی 1 را برداند ← دلت chopstick را دارد ← می تواند  
 غذا بخورد ← در لاک نذر X (باید به مستقر باشند و یکی ایی cycle گسیه  
 حالت دوم: نذر B زودتر chopstick 2 را بردارد ← چون دلت راک  
 است قبلی صفا 3 را برداند ← می تواند غذا بخورد ← در لاک نذر  
 اگر یک فیلسوف دلت راک به چید یک فیلسوف دلت چید بسیند



اگر در لاک دلتا باشد ← باید همی chopstick ها hold نباشند ← A یا  
 B باید ایی chopstick ای که بین توخ هست را hold کرده باشد ← نذر  
 یعنی صبح chopstick ای hold نکرد چون اوئی که پیش بود را نذر زودتر  
 گرفته ← به کدام از ایی دو نذر باید مستقر باشند تا اول بتواند ایی chopstick را بردارد

بتوانند بره لداغ chopstick - روشی

با

اگر  $n$  فیلسوف داشته باشیم و  $n$  chopstick و  $n$  chopstick

داشتند وی با فیلسوف هست که هیچ chopstick ای را نگیرد

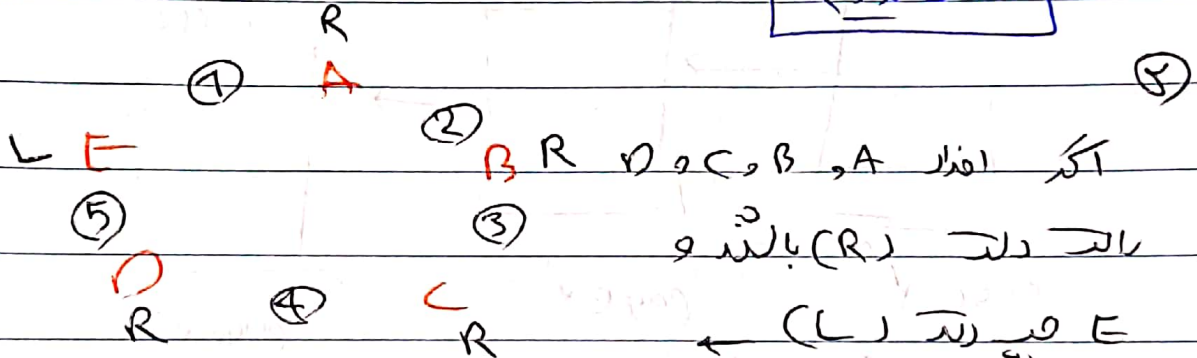
اگر  $n$  فیلسوف و  $n$  chopstick در اختیارشون است

حالا فیلسوف هست که  $n$  chopstick در اختیارش است (طبق اصل رازنه کورتی)

پس با فیلسوفی هست که درامه بخور و قوی حالت waiting

نمیست waiting cycle از بین میخ (خفن حرف باطل)

در لاک قرار



فون حدتلاب R و حدتلاب L دارم - حداقل یک R و L باید کنار هم باشند

اگر A و 2 را برداره و شروع به غذا خوردن کنه و C هم و

3 و 4 برداره و شروع به غذا خوردن کنه - E و D می توانند 5 را بردارند

تا وقتی که 6 و 4 آزارشونند - اگر اول A دلت از غذا خوردن بدست

6 و 2 آزارشونند - E و 1 را برداره و B هم 2 را برداره

وقتی C دلت از غذا خوردن بدست - 3 و 4 آزارشونند - B و 5

را برداره و 5 هم 4 را برداره - B می تواند ابدالت و غذا بخوره

ازان بعد D و 4 را برداره و بعد E و 1 را برداره و 5 که بهشون

است آزار است - اگر اول E برداره که غذاشوی خوره و بهش که

5 را آزار کرد و D هم می تواند بهش راه و غذا بخوره - هم E غذا خوره

هم D - اگر اول D و 5 را برداره هم بهش طوری - اول

ARSH D غذا بخوره بعد که 5 آزارش E می خوره - A و B و C و D و E ابدالت و غذا خوردن



انیا که سنی ندایم چون ع تا وقتی که ① را کشیده می تواند ⑤ را بکشد  
و D هم تا وقتی ④ را کشیده می تواند ⑤ را بکشد  
هر موقع یکی از این دو غذا بخورند و بعدش ⑤ را از روی کف دست بردارند و تغذیه بعدی  
می توانند شروع به غذا خوردن کنند  
حالت ریه B

اگر A و 1 را بکشد و B و 2 را و C و 3 را و D و 4 را  
E می تواند ابرای خود تا وقتی که ① از روی ⑤ را بکشد  
و D می تواند بار 5 را بکشد و غذا بخورد و بعدش 4 و 5 از روی  
می تواند C که ③ دستش بوده صیاد ④ را هم صیاد و غذا می خورد  
بعدش 3 و 4 از روی ③ را B و 2 را و 3 را هم به صیاد  
خدا می خورد و 2 و 3 از روی ③ را A و 1 را و الله و هم می گیرد  
خدا می خورد و 1 و 2 از روی ③ را E بعد از ابرای A می تواند ①  
را بکشد و اگر ⑤ از روی ⑤ را بکشد اگر از روی ⑤ صیاد کند تا D و  
5 را از روی ⑤ را بکشد E هم ابرای ⑤ را صیاد و اقلید بار ابرای ③  
هر دلی که بکشد و صیاد یک بار ابرای ⑤ را چون حادیه R و یک L و  
کار هم می افتند و تا وقتی که ریه ای به ریه یا صیاد را کشند و  
می توانند غذا بخورند و چون زمان غذا خوردن ها هم حدود 2 و 3  
حالتی ایجاب می شود که ⑤ بتوانند غذا بخورند

logical Address space =  $2^{32}$

سوال ۲

page size = 7KB =  $2^{10}$

هر رکورد  $\Rightarrow 2^2$  Ebyk  $\rightarrow 2^2$

رکورد های موجود در فایل

$$\text{تعداد page ها} = \frac{\text{حجم فضای آدرس ها}}{\text{حجم page}} = \frac{2^{32}}{2^{10}} = 2^{22} \text{ page}$$

تعداد رکورد ها در هر page  $\times$  تعداد page ها = تعداد رکورد ها در کل

$$2^2 \times 2^{22} = 2^{24}$$

بسیار بزرگ level ۱ است. باید page table را به ۲ level تقسیم کنیم.  $2^{10}$  باینری

$$\frac{2^{24}}{2^{10}} = 2^{14}$$

page table =  $2^4 \times 2^4 = 2^{16}$

اندازه page table هم ۱۶ باینری است. باید به ۲ level تقسیم کنیم.

$$\frac{2^{16}}{2^{10}} = 2^6 \rightarrow \text{page table} = 2^6 \times 2^4 = 2^{10}$$

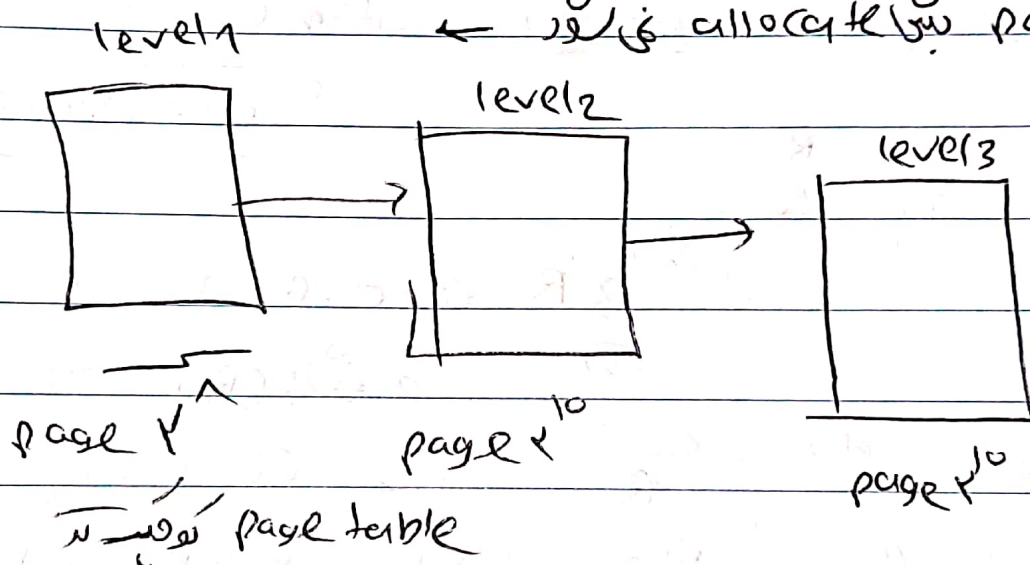
تعداد page table که از  $2^{10}$  باینری است.  $2^6$  (۶ باینری) نیاز است.

تعداد page table که از  $2^{10}$  باینری است.  $2^6$  (۶ باینری) نیاز است.  $2^{10}$  باینری (۱۰KB)  $2^6$  (۶ باینری)  $2^{10}$  باینری (۱۰KB)



2) اگر آدرسی دارد  $valid$  باشد (یعنی بین  $base + limit$  باشد) مقدار صفاتی که التماس می شود در حالت تقارن می کند چون بهی (باشد) آدرسی با  $base$  در صفحه  $level$  = ضرب مقدار  $page$  با در هر  $level$  دفعه  $2^0$

اما اگر آدرسی  $invalid$  باشد (page table entry)  $page$  -  $table$  entry  $invalid$  باشد) → ابتدا به  $level$  های بعدی می بیند و صفحه  $page$  ای از  $page$  table بسط  $allocate$  می شود ←



اگر در  $page$  table کوکرت با برای سلسله  $table$  → مقدار صفاتی  $page$  مصرف می شود اگر آدرس  $invalid$  باشد → در  $level$  های بعدی بررسی می شود.

الف) لوایح به ۴ بیت مساوی و هر بیت ۸ بیت فضای ۱ کبایت

$$1 \text{ page} = 4 \text{ KB} = 2^{12}$$

ب) max اندازه هر segment

هر segment ۶ تا page دارد

$$6 \times 2^{12} = 2^{14} \text{ byte}$$

ج) حداکثر اندازه فضای حافظه منطبق برای این سیستم

۴ segment داریم = اندازه هر segment

$$4 \times 2^{14} = 2^{16} \text{ byte}$$

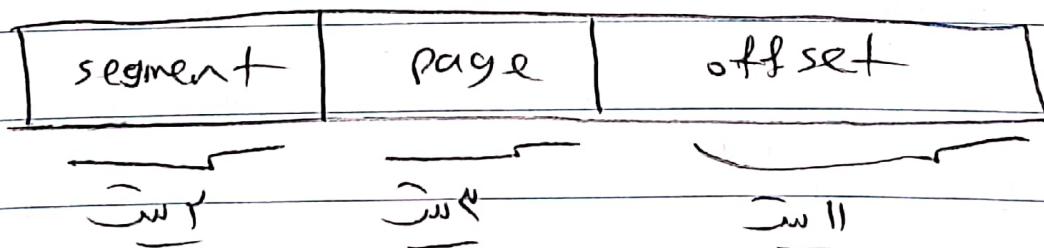
۲) آدرسی 000 27ABC

فرض آدرس منطبق و چون ۴ تا segment داریم به ۲ بیت برای ذخیره کردن

حداکثر اندازه فضای فیزیکی و نیاز است و چون ۶ تا page داریم در هر

segment ۲ بیت نیاز است

چون  $2^{12} = 4096$  page = ۱۱ بیت نیاز است



فرض آدرسی  
منطبق

۱۶ بیت در فضای  
حافظه منطبق

ب) برای ذخیره هر رقم هند ۴ بیت در باینری باید استفاده کرد

$$8 \times 4 = 32 \text{ بیت} \rightarrow \text{آدرسی ۳۲ بیت} \rightarrow \text{حداکثر اندازه} = 2^{32}$$

physical space