

به نام خدا

محمد مهدی آقا جانی

تمرین سری دوم

سیستم عامل

استاد طاهری جوان

تمرین اول : درباره فرمول Little پیرامون رابطه پایداری سیستم با تعداد فرآیندها و زمان سرویس فرآیندهای سیستم تحقیق کنید.

فرمول لیتل به صورت زیر است :

$$n = \lambda * w$$

که n در واقع برابر است با میانگین طول صف از فرآیندها و w برابر است میانگین زمان انتظار برای هر فرآیند و λ برابر نرخ ورود فرآیند به صف انتظار است . با این فرمول میتوانیم میزان پایداری را تخمین بزنیم و همچنین اگر دوتا از متغیرها را داشتیم دیگری را به دست بیاوریم

تمرین دوم : در مورد پدیده وارونگی اولویت تحقیق کنید

این پدیده در سیستم های اولویت بندی اتفاق می افتد که بیش از دو نوع اولویت دارند فرض کنید سه فرآیند داشته باشیم و فرآیند با اولویت کمتر در حال کار با منبع خاصی است حال اگر فرآیند با اولویت بالا بخواهد به منبع دسترسی داشته باشد باید منتظر اتمام فرآیند کم اولویت باشد اگر این فرآیند کم اولویت فرآیند با اولویت متوسط را صدا کند حال میزان انتظار فرآیند با اولویت بالا به زمان اتمام فرآیند با اولویت متوسط دارد که به این پدیده وارونگی اولویت میگویند

تمرین سوم: درباره شرط پاسخ گویی پردازنده به وقایع در سیستم های بلادرنگ تحقیق کنید

معیار balanced utilization در سیستم های بلادرنگ سخت که هر کار دقیقا مهلت زمانی

خاصی دارد خیلی مهم است

معیار predictability که در سیستم های بلادرنگ نرم مورد بحث است و هدف در آنجا پرهیز از

کم شدن کیفیت در سیستم ها ست.

تمرین چهارم: درباره اهداف روش های زمانبندی در سیستم های مختلف تحقیق کنید.

به طور کلی روش های زمانبندی قصد دارند به اهداف زیر برسند :

- ۱- کم کردن هزینه های سیستمی
- ۲- کاهش بار سیستم به طور منظم
- ۳- ارجحیت دادن به برنامه هایی که منابع کلیدی دارند
- ۴- استفاده متوازن از منابع سیستم
- ۵- عادل بودن

تمرین پنجم : درباره فراخوان سیستمی fork در یونیکس تحقیق کنید

وقتی یک فرآیند میخواهد برنامه دیگری را اجرا کند این کار را نمیتواند مستقیماً انجام دهد در ابتدا از خودش یک کپی ایجاد میکند که این کار از طریق فراخوانی fork انجام میشود . این فرآیند جدید در همه چیز کنترل بلاک با فرآیند پدر خود یکسان است به غیر از آی دی فرآیند سپس آن فرآیند فرزند توسط دستور exec برنامه دیگر را میتواند اجرا کند.

تمرین ششم : درباره الگوریتم زمانبندی CPU در Linux و windows ۲۰۰۰ تحقیق کنید.

نحوه اجرای فرآیندها بر روی سیستم لینوکس به صورت چندبرنامگی – تک نخه است یعنی تخصیص منابع و زمان بندی بر روی یک فرآیند انجام میشود اما در windows ۲۰۰۰ به صورت چند برنامگی چند نخه است و منابع به فرآیندها تخصیص می یابد ولی زمان بندی پردازنده بر روی thread ها صورت میگیرد.

تمرین هفتم : فرض کنید از یک الگوریتم زمانبندی ابداعی استفاده می کنیم به این صورت که فرآیندهایی برای اجرا انتخاب می شوند که در گذشته اخیر، کمترین استفاده را از زمان پردازنده داشته اند، در مورد رفتار این الگوریتم در قبال فرآیندهای با عطش I/O و فرآیندهای با عطش پردازنده بحث کنید

این الگوریتم بیشتر فرآیندهایی را سرویس میدهد که عطش I/O دارند زیرا این فرآیندها در هر بار کمتر از پردازنده استفاده میکنند در نتیجه در دفعات بعد یهم پردازنده را مال خود میکنند تا با فرآیندهایی که عطش CPU دارند برابر شود سپس پردازنده به فرآیندهای با عطش CPU تخصیص داده میشود که کوتاه مدت است و دوباره برای زمان طولانی پردازنده در اختیار فرآیندهای با عطش I/O قرار میگیرد.

تمرین هشتم : درباره انواع زمانبندی سیستم های بلادرنگ تحقیق کنید

الگوریتم های زمان بندی در سیستم های بلادرنگ میتوانند به دو صورت پویا و ایستا باشند . در حالت پویا تصمیمات زمان بندی در حین اجرا گرفته میشود و در حالت ایستا تصمیمات زمان بندی قبل از اجرا گرفته میشود . انواع زمان بندی پویا به صورت زیر است :

- ۱- Rate monotonic : در این الگوریتم هر فرآیند اولویتی بر اساس فرکانس وقوع رخداد میگیرد و فرآیند با بالاترین اولویت زودتر پردازنده را رد اختیار میگیرد.
- ۲- Earliest deadline first : لیستی از فرآیند ها بر اساس مهلتشان مرتب میشود و هر کس مهلت کمتری داشته باشد پردازنده را در اختیار میگیرد . مهلت عبارت است از زمانی که پاسخ از آن لحظه به بعد دیگر ارزشی ندارد.
- ۳- Least laxity first : این الگوریتم فرآیندی را انتخاب میکند که کمترین زمان لختی را داشته باشد . زمان لختی عبارت است از مدت زمانی که میتوان یک فرآیند را به تعویق انداخت ولی جوابش بی ارزش نشود.

تمرین نهم : درباره مفهوم Termination Cascading تحقیق کنید

فرآیندها میتوانند برای انجام کارهایشان فرآیندهای دیگر را به طریقی صدا کنند و دوباره همان فرآیندها هم میتوانند همین کار را انجام دهند در نتیجه سلسله مراتبی از فرآیندهای فرزند و پدر ایجاد میشود . در این حالت ختم فرآیند پدر میتواند مشروط به اتمام فرآیندهای فرزند باشد که به این روش cascading termination میگویند.

تمرین امتیازی یک

۱- تفاوت بین تغییر مد و تغییر فرآیند در چیست ؟

جواب : تغییر مد به آن معناست که پردازنده از یک حالت کاربر یا هسته به دیگری برود که حاصل یک وقفه است و لزوماً با تغییر فرآیند همراه نیست اما تغییر فرآیند بدان معنی است که سیستم عامل پردازنده را از یک فرآیند گرفته و به دیگری میدهد.

۲- اگر یک سیستم n پردازنده داشته باشد حداقل و حداکثر تعداد فرآیندهایی که میتواند

در حالت running یا block یا ready قرار بگیرد چقدر است ؟

جواب : در حالت اجرا تعداد فرآیندها حداکثر به تعداد پردازنده ها میباشد. اما در دو

حالت دیگری هیچ محدودیتی برای حضور فرآیندها از سمت پردازنده وجود

ندارد. برای حداقل هم میتوان گفت که صفر میتواند باشد زیرا میتوانند همه فرآیندها

منتظر I/O باشند و هیچ فرآیندی در حالت اجرا یا آماده نباشد و یا اینکه همه در حال

اجرا و آماده باشند و هیچ کس مسدود نباشد.

۳- در سیستمی که از روش MLFQ استفاده میکند مزیت استفاده از بازه های زمانی

متفاوت در صف های مختلف در چیست ؟

جواب : اول اینکه از گرسنگی بیش از حد فرآیند های طولانی جلوگیری میکند (صد در صد برطرف نمیشود) دوم اینکه با توجه به سربار بودن زمان context switching میتوان با متفاوت کردن بازه های زمانی برای صف ها این زمان سربار را کاهش داد

تمرین امتیازی دوم :

۱- هر يك از الگوريتم های زیر به چه ميزانی به فرآیند های CPU bounded اهمیت میدهند :

الف) FCFS ب) SJF ج) SRTF د) RR ه) MLFQ

جواب: الگوريتم FCFS به فرآیند های CPU bounded اهمیت بیشتری میدهد زیرا این فرآیند ها هر وقت که پردازنده را در اختیار گرفتند میتوانند کل زمان تخصیص داده شده را استفاده کنند اما فعالیت هایی که I/O bounded هستند بعد از صرف مدتی بلوکه شده و به ته صف میروند. بین SJF و SRTF اولی در آغاز بهتر از دومی به همان دلیل بالا کار میکند و برای RR هم برای اینگونه فرآیند ها ارجحیتی قایل نمیشود اما میتواند به ضرر I/O bounded ها باشد و MLFQ هم میتواند برای I/O bounded ها ارجحیت قایل شود

- در صورتی که چهار پردازنده A و B و C و D به همین ترتیب در لیست پردازنده های آماده اجرای یک زمانبند قرار داشته باشند، زمان اجرای تخمینی آنها به ترتیب برابر ۲۰ و ۴۰ و ۵۰ و ۳۰ میلی ثانیه باشد و زمان هر Switch Context بین پردازنده ها برابر ۵ میلی ثانیه باشد و از روش RR با برش زمانی ۲۰ میلی ثانیه استفاده شود، متوسط زمان پاسخگویی و متوسط زمان انتظار پردازنده ها چقدر است؟

زمان پاسخ	زمان انتظار	زمان خروج	CBT	فرایند
۱۲۰	۸۰	۱۲۰	۴۰	A
۴۵	۲۵	۴۵	۲۰	B
۱۷۵	۱۲۵	۱۷۵	۵۰	C
۱۶۰	۱۳۰	۱۶۰	۳۰	D

میانگین زمان انتظار:

$$\frac{۸۰ + ۲۵ + ۱۲۵ + ۱۳۰}{۴} = ۹۰$$

میانگین زمان پاسخ:

$$\frac{۱۲۰ + ۴۵ + ۱۷۵ + ۱۶۰}{۴} = ۱۲۵$$

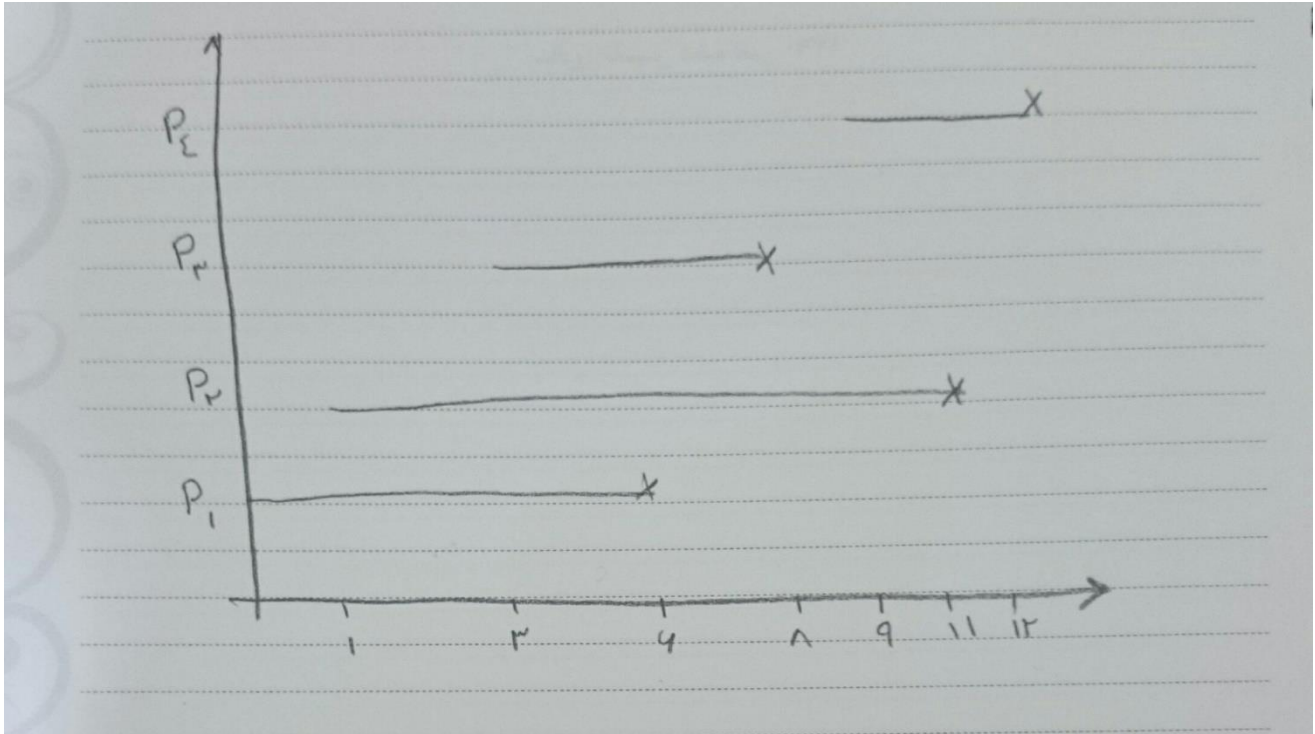
- چهار فرآیند مطابق جدول زیر در سیستم وجود دارند، اگر از روش RR با برش زمانی یک میلی ثانیه استفاده شود و از سربار تعویض متن صرف نظر شود، میانگین زمان انتظار فرآیند ها چقدر است؟

برای حل این سوال از گانت چارت استفاده میکنیم. میدانیم برش های زمانی ۱ میلی ثانیه و CBT ها به دقیقه هستند ،پس باید زمان ها را مناسب روی محور در نظر بگیریم. در ابتدای کار فقط فرآیند اول در سیستم است و تمام یک دقیقه CPU در اختیارش ست. با ورود دومین فرآیند از دقیقه اول تا سوم پردازنده بین این دو فرآیند تقسیم میشود. چون تا آمدن سومین فرآیند دودقیقه فرصت باقیست در این دو دقیقه این دو فرآیند پشت سر هم برش های یک میلی ثانیه ای پردازنده را میگیرند. در پایان دقیقه سه هر فرآیند یک دقیقه از کارش انجام شده است و به همین ترتیب جلو می رود. در نمودار پایین ریز نحوه کارها آمده است:

زمان انتظار	خروج	ورود	CBT	فرآیند
۳	۶	۰	۳	P1
۵	۱۱	۱	۵	P2
۳	۸	۳	۲	P3
۱	۱۲	۹	۲	P4

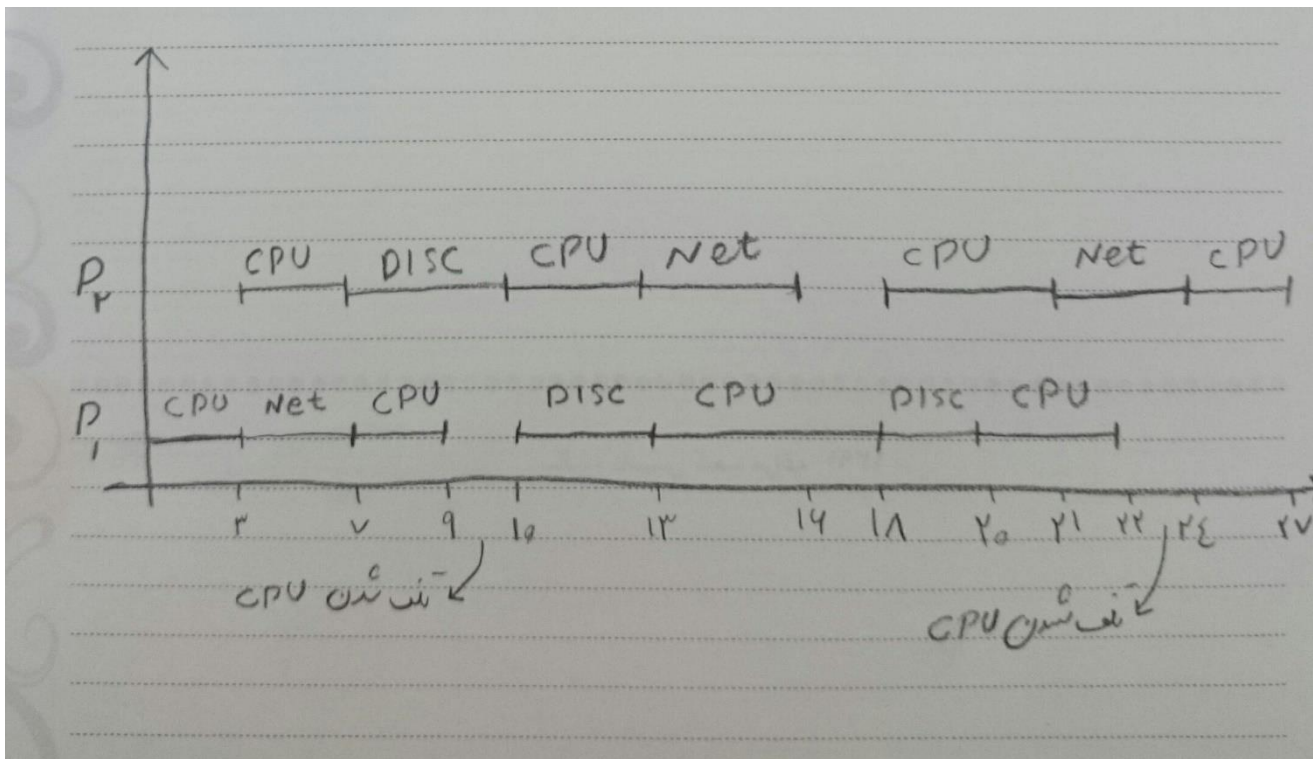
متوسط زمان انتظار:

$$\frac{3 + 5 + 3 + 1}{4} = 3$$



- دو فرآیند P_1 و P_2 با مشخصات اجرای زیر در سیستم موجودند، اطلاعات هر سطر منبع مورد نیاز برای هر پراسس و زمان مورد نیاز را مشخص می کند. مثال net^3 در سطر چهارم بیانگر این است که پراسس دوم کارت شبکه را به مدت ۳ ثانیه نیاز دارد. اگر پراسس P_2 دقیقاً ۲ ثانیه بعد از P_1 به سیستم رسیده باشد، و سیستم سیاست SJF با خاصیت Preemption را برای برنامه ریزی پراسس ها اعمال کند، زمان اجرای دو پراسس مذکور و زمان هدر رفتگی وقت CPU بر حسب ثانیه چقدر است؟

در این سوال دو بار اتلاف پردازنده داریم یعنی فرآیند اول کارش با CPU تمام شده و باید CPU به فرآیند دوم اختصاص یابد اما فرآیند دوم خودش در حال استفاده از یک منبع دیگر است. در شکل پایین زمان بندی فرآیند ها آمده است:



- یک سیستم تک پردازنده ای از الگوریتم SRT استفاده می کند. چهار فرآیند با زمان اجرای تخمینی ۶ و ۳ و ۲ و ۳ میلی ثانیه وارد سیستم می شوند. اگر زمان تعویض متن ناچیز باشد و تمامی فرآیندها فقط کار پردازشی داشته باشند، نگاه میانگین زمان انتظار فرآیندها چند میلی ثانیه است؟

به نظر میرسد به علت نداشتن زمان ورود فرایندها سوال ناقص باشد . با اطلاعات مساله اگر فرض کنیم همه فرایندها در لحظه صفر و پشت سر هم وارد شده اند و طبق الگوریتم گفته شده، میانگین زمان انتظار ۳/۷۵ است که در گزینه ها نیست.

زمان انتظار	خروج	CBT	فرایند
۸	۱۴	۶	A
۵	۸	۳	B
۰	۲	۲	C
۲	۵	۳	D

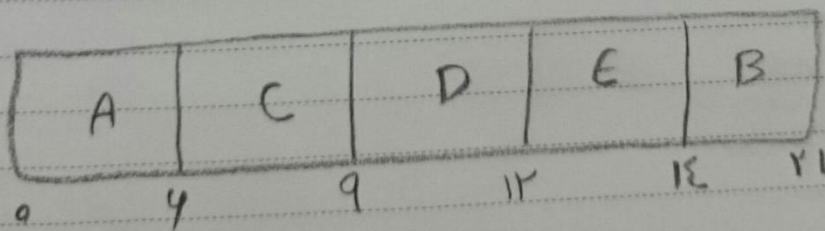
- یک سیستم تک پردازنده ای از الگوریتم زمانبندی HRRN استفاده می کند. در صورتی که فرآیندهایی با زمان اجرای ۶ و ۷ و ۳ و ۳ و ۲ به ترتیب در زمانهای ۰ و ۱ و ۳ و ۵ و ۷ وارد سیستم شوند. کدام فرآیند در انتها اجرا می شود؟

در زمان صفر چون غیر از فرایند اول چیز دیگری نداریم و اینکه الگوریتم انحصاری است پس تا لحظه ۶ فرایند اول انجام میشود. در این الگوریتم ، اولویت اجرا در لحظه تصمیم گیری باید محاسبه شود. اولویت در هر لحظه از این رابطه حاصل می شود:

زمان انتظار + زمان اجرا

زمان اجرا

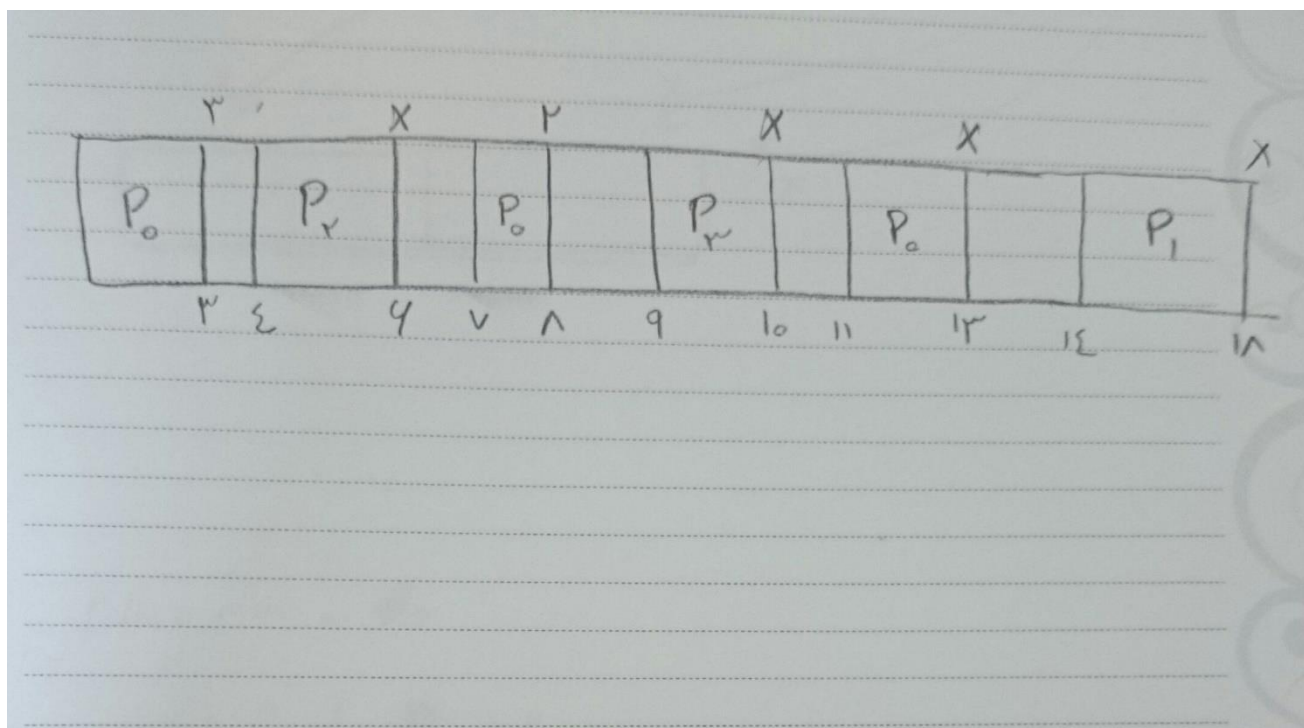
در نتیجه دومین فرآیند دیر تر از همه اجرا میشود.



- سیستمی از الگوریتم SRT استفاده می کند. اگر زمان تعویض فرآیندها امیلی ثانیه باشد، متوسط زمان انتظار برای اجرای فرآیندها چقدر است؟

فرایند	ورود	CBT	خروج	زمان انتظار
P0	۰	۶	۱۳	۷
P1	۲	۴	۱۸	۱۲
P2	۳	۲	۶	۱
P3	۸	۱	۱۰	۱

متوسط زمان انتظار برابر است با : ۵/۲۵



- در سیستمی به طور متوسط در هر ۵۰ ثانیه، ۱۰ فرآیند وارد می شوند. به طوری که میانگین زمان سرویس هر فرآیند، ۲ ثانیه است. اگر زمانبندی از نوع RR با کوانتوم زمانی ۱۰۰ میلی ثانیه باشد و زمان تعویض فرآیند ناچیز باشد، میانگین بار سیستم کدام است؟

زمان سرویس هر فرآیند ۲ ثانیه است در نتیجه برای ۱۰ فرآیند ۲۰ ثانیه زمان میشود پس ۳۰ ثانیه تلف میشود و نرخ بهره وری برابر ۴۰ درصد خواهد بود

- چهار فرآیند در یک لحظه در صف آماده قرار دارند. زمان تخمین اجرای آنها به ترتیب از P_1 تا P_4 برابر x و 7 و 5 و 8 می باشد. چنانچه حداقل میانگین زمان پاسخ فرآیندها برابر 13.5 باشد مقدار x را بیابید. (فرض بر این است که $x < 5$ است)

پاسخ	خروج	CBT	فرآیند
x	x	x	P_1
$x+12$	$x+7+5$	7	P_2
$x+5$	$x+5$	5	P_3
$x+20$	$x+5+8+7$	8	P_4

چون x کوچکتر از 5 است در ابتدا فرآیند P_1 انجام میشود و سپس P_3 و P_2 و P_4 .

$$\frac{x + x + 12 + x + 5 + x + 20}{4} = 13.5$$

در نتیجه x خواهد شد 4.25