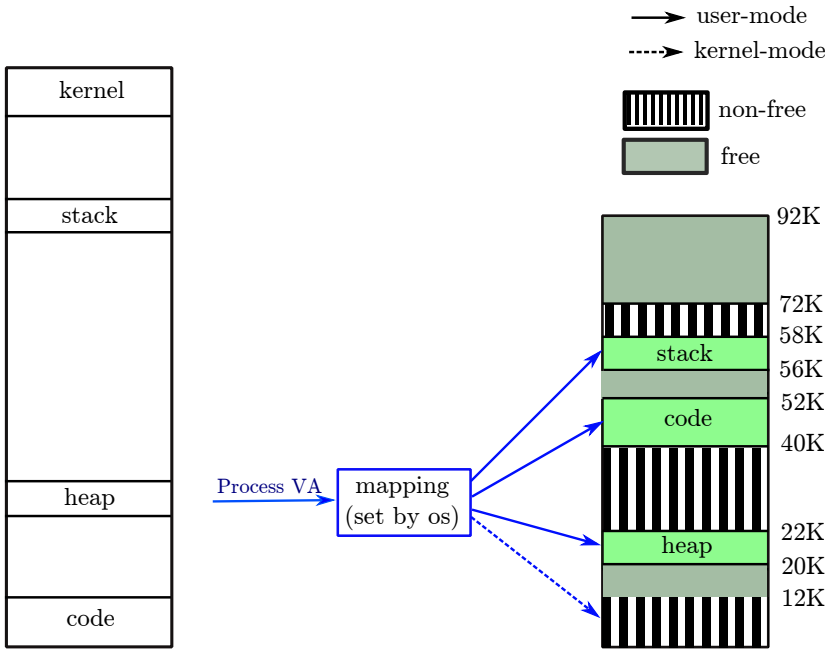
باسمه تعالی

دانشگاه صنعتی اصفهان

سیستم های عامل ـ تمرین سوم

سوالات تئوری

سوال ۱: در یک سیستم عامل فضای آدرس‌دهی هر پردازه ۱۷ بیتی (۱۲۸ کیلو بایتی) است و از روش قطعه‌سازی (segmentation) برای مجازی سازی حافظه استفاده می‌کند (تعداد قطعات برابر با ۴ است). شکل زیر وضعیت حافظه اصلی و همچنین فضای آدرس دهی یک پردازه مشخص را در یک زمان به خصوص نشان می‌دهد (مقیاس‌بندی دقیق بلوک‌های نشان داده شده بر حسب اندازه آنها مد نظر نبوده است).



الف) جدول مربوط به مدیریت حافظه شامل آدرس‌های پایه و اندازه، عملیات مجاز هنگام دسترسی و نحوه رشد هر یک از قطعات code ، stack و heap را تعیین نمایید.

ب) آدرس‌های زیر که توسط پردازه تولید می‌شود به چه آدرس‌های فیزیکی ترجمه می‌شود؟

0x100

0x17810

ج) فرض کنید هر یک از قسمت‌های stack و heap می‌توانند تا 16 KB رشد کنند (البته با این فرض که حافظه ثانویه نداریم می‌بایست در حافظه اصلی امکان اختصاص فضای بیشتر وجود داشته باشد و گرنه پردازه خاتمه می‌یابد). در این صورت اگر برنامه به ترتیب قصد استفاده از آدرس‌های زیر را داشته باشد چه اتفاقی می‌افتد؟ ضمن توضیح اگر در هر مورد مقادیری از جدول مدیریت حافظه تغییر می‌کند آنها را مشخص کنید (به ازای هر مورد حداکثر دو تغییر مقدار مجاز است).

0x14800
0xb000

سوال ۲:

فرض کنید که فضای آدرس دهی مجازی ۱۵ بیتی است و اندازه هر صفحه ۳۲ بایت است. همچنین فرض کنید که از روش paging با ۲ سطح استفاده میکنیم، یعنی یک page directory داریم که به صفحات مختلف page table اشاره می کند. هر مولفه page directory (یا همان PDE) یک بایت است. هر PDE را که در نظر بگیریم، پر ارزشترین بیت (سمت چپ ترین بیت)، به عنوان بیت اعتبار (valid bit) در نظر گرفته می شود و سایر بیتها (۷ بیت)، PFN صفحه ای از page table را مشخص می کنند. مولفه های page table (یا همان PTE ها) ساختار مشابهی دارند. یعنی هر PTE نیز یک بایت است که پر ارزشترین بیت، بیت اعتبار بوده و ۷ بیت مابقی PFN صفحه مورد نظر (صفحه ای که می خواهیم نهایتا به آن دسترسی پیدا کنیم) را مشخص می کنند.

فرض کنید که page directory در صفحه ای با شماره فیزیکی ۲۰ قرار دارد (PFN=20).

محتوای صفحات فیزیکی زیر در اختیار شما قرار داده شده است (محتوای هر بایت در فرمت هگز نشان داده شده است).

Page with PFN=5	8b 66 10 45 af e7 3c cb b1 a7 31 d2 2a e8 b2 65 f8 a0 cb 18 58 66 24 0e ae 2e c9 74 c3 1c ab a6
Page with PFN=9	6d 5d cd 8a f7 0d a5 f3 a9 62 06 2b 5d b9 2f 4e ce ba b5 a6 a0 ec bd 69 98 43 73 cf 08 1c f1 35
Page with PFN=20	f0 ac 8e ea 23 20 19 f5 6f fa fc 5a 34 a8 89 c9 bf 5f 5e 76 32 4b cb 0d ec 06 bb 34 a7 93 81 89
Page with PFN=58	2d ce 12 17 3e 10 8c 42 53 8e a1 60 86 fc 7e 26 ee b9 5e 1b ad 6b d5 8f 85 95 c6 7b f5 db f0 5a
Page with PFN=105	b9 9a 05 73 bd 4b 2d 29 5c 87 e5 7f 0f 3a 9a 43 c0 1f a3 61 44 52 51 e1 e6 05 bc 8a e0 40 3b 59
Page with PFN=127	be 91 ec 48 fb d6 78 4b 8a 42 75 cc a2 08 4f f4 ce 9e f8 b9 07 a8 5e dd 59 de a3 a1 6a ec 7c 6f

به سوالات زیر پاسخ دهید:

الف) فرض کنید فرآیندی میخواهد بایتی که در آدرس مجازی 0x3a3b قرار دارد را بخواند.

الف_۱) برای ترجمه آدرس مجازی به آدرس فیزیکی به چه صفحاتی می بایست دسترسی پیدا شود؟جواب خود را توضیح دهید.
الف_۲) آیا این آدرس مجازی به این فرآیند تخصیص داده شده است؟ اگر «خیر» چرا؟ و اگر «بله»، بایت مورد نظر که قصد خواندن آن را داریم در چه صفحه ای قرار دارد و مقدار آن چیست؟ (توضیح دهید.)
ب) قسمت قبل را برای آدرس مجازی 0x303b تکرار کنید.

سوال ۳:
یک سیستم مدیریت حافظه از نوع سگمنتی صفحه بندی شده (paged segmentation) را در نظر بگیرید. همچنین فرض کنید در این سیستم مدیریت حافظه اندازه هر سگمنت بتواند حداکثر تا ۴ گیگابایت رشد کند و اندازه هر رکورد جدول صفحه ۴ بایت باشد. برای آنکه متوسط سربار (مجموع فضای لازم برای ذخیره جداول و قطعه شدگی داخلی [۱]) این سیستم مدیریت حافظه حداقل شود اندازه صفحات می‌بایست چند بایت باشد؟
راهنمایی: برای محاسبه مقدار متوسط سربار فرض کنید اندازه هر سگمنت عددی تصادفی بین ۰ و ۴ گیگابایت است. همچنین به دلیل مدیریت خوب حافظه توسط پردازه‌ها، در سگمنت‌های از نوع heap نیز فضای خالی قابل ملاحظه‌ای ایجاد نمی‌شود.

سوال ۴:
قطعه کد زیر را در نظر بگیرید:

```
int a[1024][1024], b[1024][1024], c[1024][1024];
multiply() {
int i, j, k;
for (i = 0; i < 1024; i++)
  for (j = 0; j < 1024; j++)
    for (k = 0; k < 1024; k++)
      c[i][j] += a[i,k] * b[k,j];
}
```

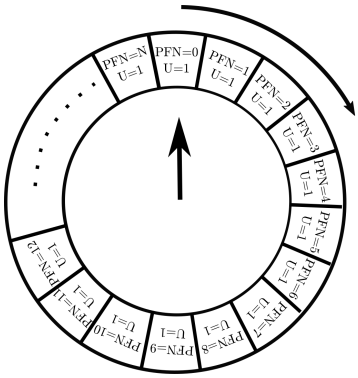
فرض کنید که فایل باینری اجرایی این برنامه در یک صفحه و قسمت پشته (stack) نیز در یک صفحه جایگذاری شده است. همچنین اندازه هر عدد صحیح را ۴ بایت و تعداد رکوردهای TLB را ۸ عدد در نظر بگیرید که همواره ترجمه شماره صفحات مجازی که نسبت به سایر صفحات جدیدتر استفاده شده‌اند را در برمی‌گیرند. در این صورت تعداد فقدان‌های TLB هنگام اجرای این برنامه چه تعداد است؟ (فرض کنید در ابتدای کار TLB خالی است)
راهنمایی: برای حل این سوال نیاز دارید که بدانید درایه‌های یک آرایه به صورت سطری و یا ستونی در خانه‌های متوالی حافظه نوشته می‌شوند. برای بررسی این موضوع می‌توانید از قطعه کد زیر استفاده نمایید:

```
#include <stdio.h>
int main ( )
{
  int x[2][2];
  x[0][0]=1;
  x[0][1]=2;
  x[1][0]=3;
  x[1][1]=4;
  int *p;
  p=x[0];
  for(int i=0;i<=3;i++)
    printf("%d\n",*(p+i));
  return (0);
}
```

[۱] Internal fragmentation

سوال ۵:

الگوریتم ساعت (clock) برای مدیریت حافظه اصلی را در نظر بگیرید. فرض کنید که وضعیت صفحات اصلی و عقربه ساعت به صورت مشخص شده در شکل زیر است. (بیت use مربوط به هر صفحه به اختصار با U نشان داده شده است و در شکل زیر این بیت برای تمام صفحات ۱ است)



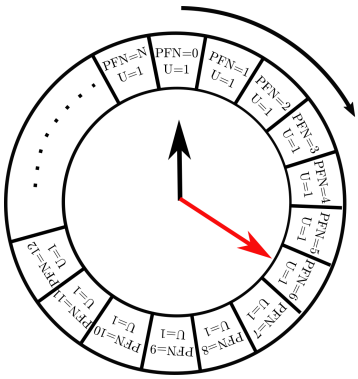
الف) در این وضعیت اگر یک درخواست برای ورود صفحه از دیسک به حافظه اصلی دریافت شود چه صفحه‌ای برای خارج شدن از حافظه اصلی انتخاب می شود؟ این عمل انتخاب چه مقدار زمان طول می کشد؟ توضیح دهید. (فرض کنید بررسی کردن بیت use هر صفحه معادل یک واحد زمانی طول می کشد)

ب) برای بهبود عمل‌کرد الگوریتم ساعت نسخه های متنوعی ارائه شده است. در زیر توصیف دو نوع از این نسخه ها آمده است:

ب ۱) الگوریتم A : در این نسخه به ازای هر صفحه یک متغیر m بیتی به نام F وجود دارد. عقربه ساعت بر روی هر صفحه که قرار می‌گیرد ابتدا مقدار F آن صفحه را با بیت U متناظر با این صفحه به صورت زیر به روز می‌کند و سپس بیت U را صفر می کند. نهایتاً در این الگوریتم صفحه‌ای برای خروج از حافظه اصلی انتخاب می‌شود که مقدار متغیر F کوچکتری دارد.

```
F = ( U << ( m - 1 )) | ( F >> 1 )
```

ب ۲) الگوریتم B : در این نسخه علاوه بر عقربه قبلی یک عقربه بزرگ‌تر هم وجود دارد. موقعیت عقربه ها نسبت به همدیگر قفل است. عقربه جدید بزرگ (عقربه جلویی) وظیفه صفر کردن بیت use صفحاتی که بر روی آن‌ها قرار می‌گیرد را بر عهده دارد و عقربه کوچک همان وظیفه قبلی خود در نسخه اولیه را انجام می دهد. شکل زیر نمایی از این الگوریتم را به صورت تصویری نشان می دهد.



به نظر شما دلیل پیشنهاد هر یک از این دو نوع الگوریتم چیست و درصدد حل چه مشکلی و یا بهبود چه پارامتری در الگوریتم ساعت اولیه هستند؟ توضیح دهید.