باسمه تعالی

تکلیف شماره دو درس سیستم عامل – استاد جوادی

سید امیرمهدی میرشریفی-۹۸۳۱۱۰۵

۱) پس از اجرای قطعه کد زیر، چند پردازه خواهیم داشت؟ به صورت مختصر توضیح دهید. (فرض کنید فراخوانی‌های سیستمی با موفقیت اجرا می‌شوند.)

```c
int main(){
    fork() && fork() || fork();
    for (int i=0; i<3; i++){
        fork();
    }
    return 0;
}
```

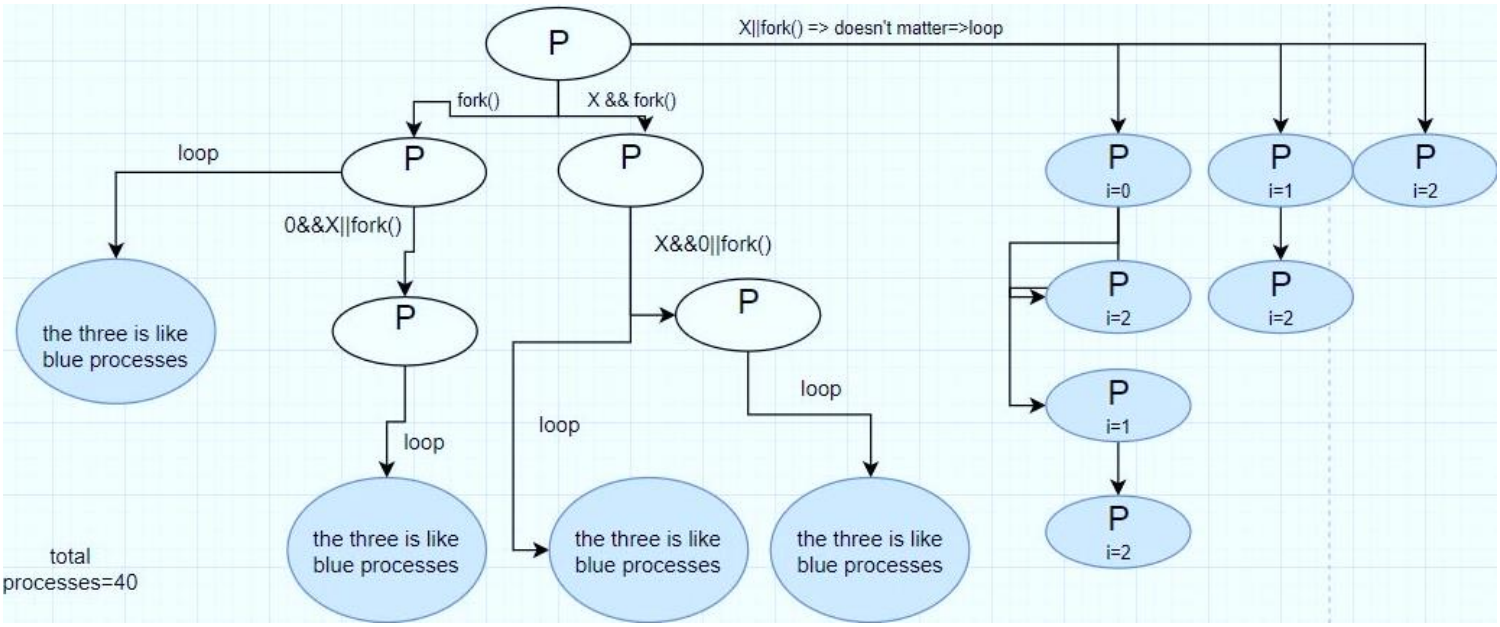
پاسخ: ۴۰ پردازه

در این سوال در نظر گرفتن چند نکته الزامی است:

در زبان C اولویت &&از || بیشتر است .

Circuiting-Shortداریم. یعنی در شرط a && b اگر a==0 باشد دیگر b چک نمیشود و در a||b اگر a==1 باشد دیگر b چک نمیشود.

در ادامه درخت آن رسم شده است. هر چند برای سادگی کار جاهایی که تکراری بود صرفا علامت گذاری شده است.(تا حد امکان دلیل ایجاد پردازه جدید مشخص شده است)



۲) تکه کد زیر از Pthread استفاده میکند. خروجی خطوط C و P چیست؟

```
#include <pthread.h>
#include <stdio.h>
#include <types.h>
int value = 0;
void *runner(void *param); /* the thread */
int main(int argc, char *argv[]) {
pid t pid;
pthread_t tid;
pthread_attr_t attr;
pid = fork();
if (pid == 0) {
pthread_attr_init(&attr);
pthread_create(&tid, &attr, runner, NULL);
pthread_join(tid, NULL);
printf("CHILD: value = %d", value); /* LINE C */
} else if (pid > 0) {
wait(NULL);
printf("PARENT: value = %d", value); /* LINE P */
}
}
void *runner(void *param) {
value = 5;
pthread_exit(0)                                                    ;
```

پاسخ:

```
CHILD: value = 5/* LINE C */
```

```
PARENT: value = 0/* LINE P */
```

۳) بیشترین و کمترین مقداری که در خروجی دیده می شود را با ذکر دلیل بیان کنید.

```
int count = 0;
interleave() {
   pthread_t th0, th1;
   pthread_create(&th0, 0, test, 0);
   pthread_create(&th1, 0, test, 0);
   pthread_join(th0, 0);
   pthread_join(th1, 0);
   printf(count);
}

test() {
   for (int j = 0; j < MAX; j++) count = count + 1;
}
```

پاسخ:

حداکثر 2*max و حداقل max.

در کد بالا دو ریسمان ساخته می شود که توابعی که آنها فراخوانی میکنند ، متغییری را **تغییر می دهد** که جزو کد اصلی است و بنابراین برای هر دو ریسمان **مشترک** است.

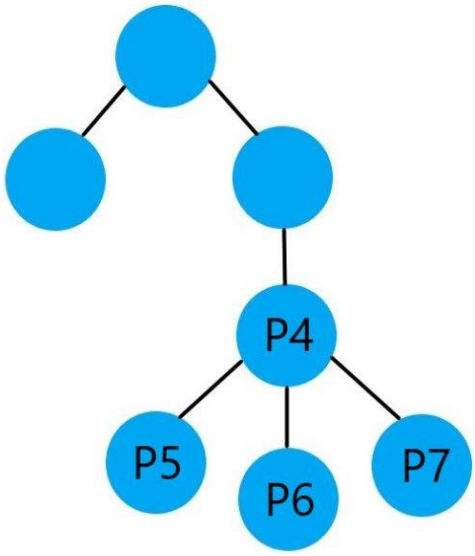
بنا براین باید احتمال race condition را در نظر گرفت.

با توجه به این که در هر دور از حلقه به متغیر ما یک واحد اضافه میشود:

۱) اگر context switch قبل از ذخیر متغیر اعمال شود و پردازه به ریسمان دیگر ما اختصاص یابد، متغیر به روز رسانی نخواهد شد و با همان مقدار قبلی به ریسمان دوم میرود. بنا بر این وقتی نوبت به ذخیره متغییر در این ریسمان ها میشود هر دو ریسمان بعد از اتمام یک دور از حلقه ، یک مقدار را ذخیره می کنند و یعنی بعد از اتمام یک دور از حلقه در دو ریسمان ، به جای این که متغییر ما دو واحد افزایش داشته باشد ، فقط یک واحد افزایش می یابد. از این رو در پایان کار توابع ریسمان ها ، متغییر ما مقداری معادل با مقدار max پیدا کرده است.

۲) اگر context switch بعد از دخیره متغیر جدید اعمال شود ، بنا براین بعد از هر دور از حلقه در ریسمان ها متغیر ما دو واحد افزایش می یابد و در آخر متغییر ما برابر با 2*max میشود.

۴) شبه کدی بنویسید که درخت فرآیند زیر را به وجود آورد، به گونه‌ای که فرآیندهای ***P5***، ***P6*** و ***P7*** برنامه‌های ***ls***، ***sort*** و ***search*** را اجرا کرده و فرآیند پدر آن‌ها (***P4***) تا زمان اتمام کار این سه فرآیند منتظر آن‌ها بماند.



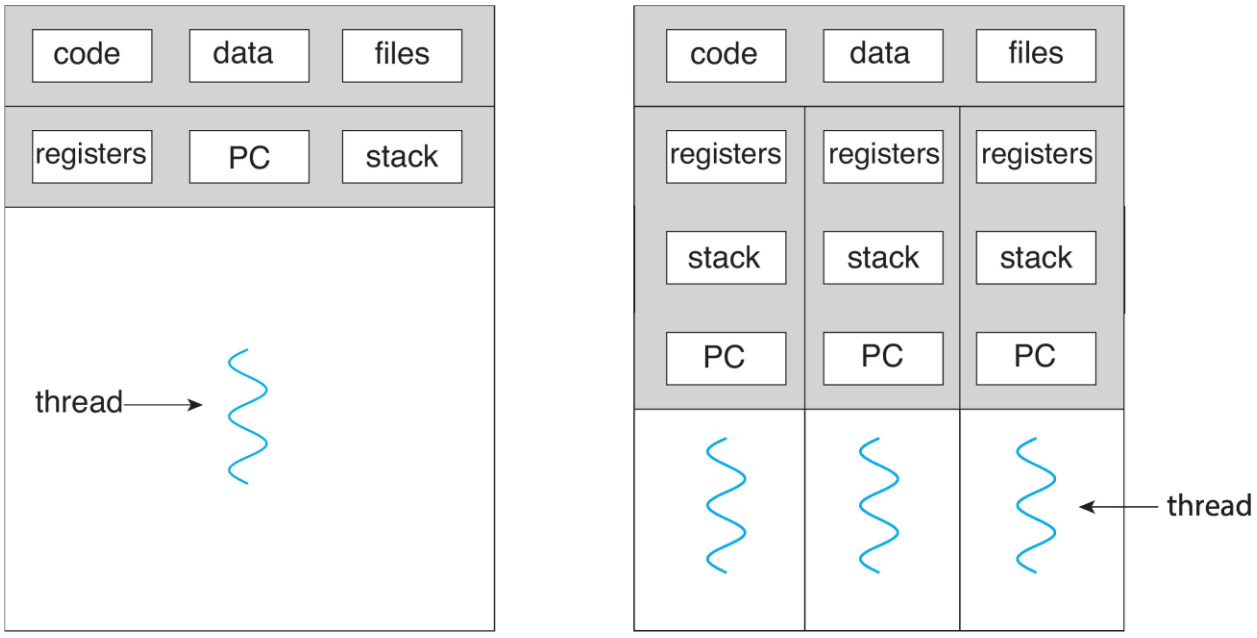
پاسخ:

```
If  fork()>0   then  /*father process*/
        If fork()==0 then
                If fork()==0 then /*P4*/
                        If fork()==0 /*P5*/ then : ls()  ; exit()
                        If fork()==0 /*P6*/ then: sort() ; exit()
                        If fork()==0 /*P7*/ then: search() ; exit()
                        wait()
                        wait()
                        wait()
```

۵) در هنگام ساخت یک ریسمان (***thread***) از چه منابعی در سیستم استفاده می‌شود و چه تفاوتی بین این منابع و منابع سیستمی مورد نیاز در زمان ساخت پردازه وجود دارد؟

هنگام ساخت یک ریسمان ، منابع خاصی در اختیار ریسمان قرار نمی گیرد و هزینه ساخت آن بسیار کمتر از هزینه ساخت یک پردازه است . از این رو است که به آن lightweight process می گویند.

در هنگام ساخت یک ریسمان منابع با پردازه ای که آن ریسمان را ایجاد کرده است مشترک است. منابعی همچون کد ها و دیتا ها و همچنین فایل ها. اما برای هر ریسمان یک استک، مجموعه رجیستر و pc مجزا در نظر گرفته میشود. این در حالی است که زمان ساخت یک پردازه جدید تمام منابع اعم از کد ها دیتا ها و فایل ها و همچنین استک و رجیستر ها و pc جدیدی برای پردازه تخصیص داده می شود.



single-threaded process

multithreaded process

سوال امتیازی

6) در هر عمل ***context switch*** میان دو ریسمان متعلق به یک پردازه، چه مواردی باید ذخیره و بازیابی شوند؟ در صورتی که این عمل میان دو پردازه انجام شود چطور؟ با توجه به پاسخ خود نتیجه‌گیری کنید که چرا در موارد زیادی استفاده از ریسمان‌ها به جای پردازه‌ها در سیستم می‌تواند سودمند باشد.

در زمان context switch میان دو ریسمان متعلق به یک پردازه، چون منابع یکسان و متعلق به پردازه است نیازی به ذخیره و تعویض آنها نیست و از این رو که تنها تفاوت میان دو ریسمان، استک و رجیستر ها و pc آنها است، بنابراین این موارد می بایست ذخیره شوند و جایشان را با استک و pc و رجیستر های ریسمان جدید عوض کنند.

برای عمل context switch بین دو پردازه تمام منابع و استک و رجیستر و pc می بایست عوض شوند و جایشان را به منابع و .. پردازه جدید دهند.

از این رو thread context switch ، سربار کمتری را نسبت به process context switch به سیستم تحمیل می کند.