

۱- با استفاده از برنامه نشان داده شده در شکل ۱، توضیح دهید که خروجی در LINE A چه خواهد بود.

```
#include <sys/types.h>
#include <stdio.h>
#include <unistd.h>

int value = 5;

int main()
{
    pid_t pid;

    pid = fork();

    if (pid == 0) { /* child process */
        value += 15;
        return 0;
    }
    else if (pid > 0) { /* parent process */
        wait(NULL);
        printf("PARENT: value = %d",value); /* LINE A */
        return 0;
    }
}
```

شکل ۱

۲- با احتساب فرآیند والد اولیه، چند فرآیند توسط برنامه نشان داده شده در شکل‌های ۲ و ۳ ایجاد می شود؟ تشریح نمایید

```
#include <stdio.h>
#include <unistd.h>

int main()
{
    /* fork a child process */
    fork();

    /* fork another child process */
    fork();

    /* and fork another */
    fork();

    return 0;
}
```

شکل ۲

```
#include <stdio.h>
#include <unistd.h>

int main()
{
    int i;

    for (i = 0; i < 4; i++)
        fork();

    return 0;
}
```

شکل ۳

۳- نسخه های اصلی سیستم عامل iOS موبایل هیچ مکانیزمی برای پردازش همزمان ارائه نمی دهند. سه مشکل اصلی را که پردازش همزمان به یک سیستم عامل اضافه می کند، بیان کرده و توضیح دهید.

۴- هنگامی که یک فرآیند با استفاده از عملیات `fork()` یک فرآیند جدید ایجاد می کند، کدام یک از موارد زیر بین فرآیند والد و فرآیند فرزند به اشتراک گذاشته می شوند؟

- a. Stack
- b. Heap
- c. Shared memory segments

۵- معنای "دقیقا یک بار" یا `exactly once` را با توجه به مکانیسم RPC در نظر بگیرید. آیا الگوریتم اجرایی این مفهوم به درستی اجرا می شود حتی اگر پیام ACK ارسال شده به کلاینت به دلیل مشکل شبکه از بین برود؟ توالی پیام ها را توصیف کنید و در مورد اینکه آیا "دقیقا یک بار" هنوز حفظ شده است یا خیر بحث کنید.

۶- فرض کنید که یک سیستم توزیع شده مستعد خرابی سرور است. چه مکانیسم هایی برای تضمین معنایی "دقیقا یک بار" برای اجرای RPC ها مورد نیاز است؟

۷- اقدامات انجام شده توسط یک هسته برای Context Switching بین فرآیندها را توضیح دهید.

۸- نقش فرآیند `init` (یا `systemd`) در سیستم های یونیکس و لینوکس را در رابطه با خاتمه فرآیند توضیح دهید.

۹- با استفاده از برنامه شکل ۴، مقادیر `pid` را در خطوط A، B، C و D شناسایی کنید.

```

#include <sys/types.h>
#include <stdio.h>
#include <unistd.h>

int main()
{
    pid_t pid, pid1;

    /* fork a child process */
    pid = fork();

    if (pid < 0) { /* error occurred */
        fprintf(stderr, "Fork Failed");
        return 1;
    }
    else if (pid == 0) { /* child process */
        pid1 = getpid();
        printf("child: pid = %d",pid); /* A */
        printf("child: pid1 = %d",pid1); /* B */
    }
    else { /* parent process */
        pid1 = getpid();
        printf("parent: pid = %d",pid); /* C */
        printf("parent: pid1 = %d",pid1); /* D */
        wait(NULL);
    }

    return 0;
}

```

شکل ۴

۱۰- با استفاده از قانون Amdahl، بهره افزایش سرعت برنامه‌ای را محاسبه کنید که ۶۰ درصد مؤلفه موازی برای (الف) دو هسته پردازشی و (ب) چهار هسته پردازشی دارد.

۱۱- دو تفاوت بین نخ‌های سطح کاربر و نخ‌های سطح هسته را بیان نمایید. در چه شرایطی یک نوع بهتر از دیگری است؟

۱۲- اقداماتی را که توسط یک کرنل برای Context Switching بین نخ‌های سطح هسته انجام می‌شود، توضیح دهید.

۱۳- تحت چه شرایطی یک راه حل چند نخ‌ی با استفاده از نخ‌های هسته چندگانه عملکرد بهتری نسبت به راه حل تک نخ‌ی در یک سیستم تک پردازنده‌ای ارائه می‌دهد؟

۱۴- فرآیندهای زیر با استفاده از یک الگوریتم زمان‌بندی غیرانحصاری RR برنامه‌ریزی می‌شوند. به هر فرآیند یک اولویت عددی اختصاص می‌یابد که عدد بالاتر نشان‌دهنده اولویت نسبی بالاتر است. علاوه بر فرآیندهای ذکر شده، سیستم دارای یک task بیکار است (که هیچ منبع CPU را مصرف نمی‌کند و به عنوان Pidle شناخته می‌شود) این task دارای اولویت صفر بوده و زمانی برنامه‌ریزی می‌شود که سیستم هیچ فرآیند دیگری برای اجرا نداشته باشد. طول کوانتوم زمانی ۱۰ واحد است. اگر فرآیندی با یک فرآیند با اولویت بالاتر از پیش گرفته شود (Preempted)، فرآیند پیش‌پرداخت (preempted process) در انتهای صف قرار می‌گیرد.

<u>Process</u>	<u>Priority</u>	<u>Burst</u>	<u>Arrival</u>
P_1	40	20	0
P_2	30	25	25
P_3	30	25	30
P_4	35	15	60
P_5	5	10	100
P_6	10	10	105

الف) ترتیب زمانبندی فرآیندها را با استفاده از نمودار گانت نشان دهید.

ب) زمان برگشت برای هر فرآیند چقدر است؟

ج) زمان انتظار برای هر فرآیند چقدر است؟

د) میزان استفاده از CPU چقدر است؟

۱۵- در مورد اینکه چگونه جفت معیارهای زمانبندی زیر در تنظیمات خاصی با هم در تضاد هستند توضیح دهید:

الف) استفاده از CPU و زمان پاسخ

ب) میانگین زمان برگشت و حداکثر زمان انتظار

ج) استفاده از دستگاه I/O و استفاده از CPU

۱۶- یک الگوریتم زمان بندی اولویت غیرانحصاری را بر اساس اولویت ها در حال تغییر در نظر بگیرید. اعداد اولویت بزرگتر به معنای اولویت بالاتر است. هنگامی که یک فرآیند منتظر CPU است (در صف آماده، اما در حال اجرا نیست)، اولویت آن با نرخ α تغییر می کند. هنگامی که در حال اجرا است، اولویت آن با نرخ β تغییر می کند. به تمام فرآیندها وقتی وارد صف آماده می شوند اولویت صفر داده می شود. پارامترهای α و β را می توان برای ارائه الگوریتم های زمان بندی مختلف تنظیم کرد.

الف) الگوریتمی که از $\beta > \alpha > 0$ حاصل می شود چیست؟

ب) الگوریتمی که از $\alpha < \beta < 0$ حاصل می شود چیست؟

موفق باشید