



سیستم‌های عامل (۴۰۴۲۴) - گروه ۲  
نیم‌سال دوم سال تحصیلی ۱۴۰۱-۱۴۰۲  
استاد درس: دکتر رسول جلیلی

تمرین اول

مهلت تحویل: ۲۳:۵۹ روز یکشنبه ۱۴۰۱/۱۲/۲۸

نکات و قواعد

۱. محل بارگذاری تمرین تا سه روز پس از مهلت تحویل باز خواهد بود. در طول ترم، برای تکالیف عملی و تئوری در مجموع می‌توانید از ۱۲ روز تاخیر مجاز به صورت ساعتی استفاده کنید و پس از آن به ازای هر روز ۲۵ درصد جریمه بر روی نمره‌ی کسب شده اعمال خواهد شد.
۲. لطفا حتماً **آداب‌نامه‌ی انجام تمرین‌های درسی** را رعایت نمایید. در صورت استفاده از هر مرجعی و یا همفکری برای پاسخ به سوالات، مرجع مربوطه و یا نام همفکران را در پاسخ خود ذکر کنید.
۳. در صورتی که پاسخ سوالات را به صورت دست‌نویس آماده کرده‌اید، لطفاً تصاویر واضحی از پاسخ‌های خود ارسال کنید. در صورت ناخوانا بودن پاسخ ارسالی، نمره‌ای به پاسخ ارسال شده تعلق نمی‌گیرد.
۴. فایل مربوط به پاسخ خود را به فرمت `OS_HW\StdNum_FirstName_LastName` نامگذاری کرده و ارسال نمایید.

سوال ۱ (۳۳ نمره)

به سوالات زیر با توضیحات کافی پاسخ دهید.

- (الف) (۱۶ نمره) دنباله گام‌هایی که با رخداد وقفه زمانی و نهایتاً تعویض بافتار<sup>۱</sup> به برنامه دیگر طی می‌شود را شرح دهید. نقش بلوک کنترل پردازش<sup>۲</sup> را در این ارتباط روشن نمایید.
- (ب) (۱۰ نمره) تغییر وضعیت یک فرآیند از حالت اجرا به انتظار به دو صورت رخ می‌دهد. دو عاملی که باعث این تغییر وضعیت می‌شوند را نام ببرید و برای هر کدام چگونگی تغییر وضعیت و عوامل دخیل را شرح دهید.
- (ج) (۷ نمره) ساختارهای یکپارچه<sup>۳</sup> و پیمانه‌ای<sup>۴</sup> برای سیستم‌های عامل را از نظر کارایی و گسترش‌پذیری مقایسه کنید.

سوال ۲ (۱۰ نمره)

نقش اعمال دو حالتی<sup>۵</sup> در سیستم عامل چیست؟ تغییر حالت به چه صورت رخ می‌دهد؟ اجزای سخت‌افزاری و نرم‌افزاری دخیل در تغییر حالت و روند این تغییر را توضیح دهید.

سوال ۳ (۱۷ نمره)

- فرآیندهای داخل یک سیستم، به دلایل مختلفی نیاز به ارتباط و تبادل اطلاعات با یکدیگر دارند. به ارتباطات بین فرآیندها، ارتباطات بین فرآیندی<sup>۶</sup> گفته می‌شود. در این مورد به سوالات زیر پاسخ دهید.
- (الف) (۷ نمره) این ارتباطات دو نوع دارند. هر یک را نام برده و عملکرد کلی هر یک را به صورت مختصر توضیح دهید.
  - (ب) (۵ نمره) فرض کنید چندین فرآیند داریم که روی سیستم‌های مختلف در داخل یک شبکه در حال اجرا هستند. در چنین حالتی، کدام نوع از ارتباط بین فرآیندی مناسب است؟ چرا؟

۱. Context Switch
۲. Process Control Block
۳. Monolithic
۴. Modular
۵. Dual-mode Operations
۶. Interprocess Communication (IPC)

(ج) (۵ نمره) مفاهیم مسدودکننده<sup>۷</sup> و غیر مسدودکننده<sup>۸</sup> را توضیح داده و ربط آن به ارتباطات بین فرآیندی را توضیح دهید.

#### سوال ۴ (۱۰ نمره)

برنامه‌ی زیر را در نظر بگیرید.

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

int main(int argc, char *argv[])
{
    pid_t pid;
    int status;
    pid = fork();
    if (pid != 0) {
        while (pid != wait(&status));
    }
    else {
        sleep(5);
        exit(5);
    }
    pid = fork();
    if (pid != 0) {
        while (pid != wait(&status));
    }
    else {
        sleep(1);
        exit(1);
    }
}
```

در این برنامه، یک فرآیند جدید ایجاد می‌شود و برنامه منتظر پایان فرآیند می‌ماند. پس از آن، یک فرآیند جدید ایجاد می‌شود و روند قبلی تکرار می‌شود. این کد را برای ایجاد دو فرآیند مختلف (که به صورت موازی اجرا می‌شوند) تغییر دهید به صورتی که فرآیند پدر منتظر خاتمه هر دوی آنها بماند.

#### سوال ۵ (۲۰ نمره)

تکه کد زیر را در نظر بگیرید.

```
int a = 5;
int fd = open(..); // opening a file
int ret = fork();

if (ret > 0) {
    close(fd);
    a = 6;
    ...
    ...
}
else if (ret == 0) {
    printf("a=%d\n", a);
    read(fd, something);
}
```

فرض کنید بعد از ایجاد فرآیند جدید با فراخوانی fork، والد قبل از فرزند نوبت اجرا در پردازنده می‌گیرد و فرآیند فرزند بعد از اجرای کامل فرآیند والد زمان‌بندی و اجرا می‌شود. با این توضیحات به سؤالات زیر پاسخ دهید:

۷. Blocking  
۸. Non-blocking

- (الف) (۵ نمره) مقدار  $a$  که در فرآیند فرزند در خروجی چاپ می‌شود چند است؟
- (ب) (۵ نمره) آیا تلاش برای خواندن از  $fd$  در فرآیند فرزند موفقیت آمیز خواهد بود؟
- (ج) (۱۰ نمره) با ترتیب زمان‌بندی که بیان شد، آیا فرآیند فرزند به حالت زامبی یا یتیم تبدیل می‌شود؟ در صورت مثبت بودن جواب، با تغییر در کد داده شده، از این اتفاق جلوگیری کنید.

### سوال ۶ (۱۰ نمره)

برای قطعه کد زیر، خروجی چاپ شده در خطوط  $LINE\ X$  و  $LINE\ Y$  را مشخص کنید.

```
#define SIZE 5

int nums[SIZE] = {0, 1, 2, 3, 4};

int main()
{
    int i;
    pid_t pid;

    pid = fork();

    if (pid == 0) {
        for (i = 0; i < SIZE; i++) {
            nums[i] *= -i;
            printf("CHILD: %d\n", nums[i]); /* LINE X */
        }
    }
    else if (pid > 0) {
        wait(NULL);
        printf("***Child Complete***\n");
        for (i = 0; i < SIZE; i++) {
            printf("PARENT: %d\n", nums[i]); /* LINE Y */
        }
    }
    return 0;
}
```

### سوال ۷ امتیازی (۱۵ نمره)

کد صفحه‌ی بعد قسمتی از پیاده‌سازی مربوط به اجرای پایپلاین دو دستور (مانند `ls | grep new`) آمده است. با استفاده از دانش خود و جستجوی دستورات و فراخوانی‌های سیستمی ناآشنا، عملکرد خطوط مشخص شده‌ی  $LINE\ ۱$  تا  $LINE\ Y$  را توضیح دهید و مشخص کنید که این کد چگونه اجرای پایپلاین دو دستور را پیاده‌سازی می‌کند.

```
void makePipeline(char* command1, char** argv1, char* command2, char** argv2) {
    int fd[2];
    pipe(fd);                                /* LINE1 */
    pid_t pidComm[2];
    pidComm[1] = fork();
    if (pidComm[1] == -1) {
        printf("Can't create new process for pipeline\n");
        return;
    }
    else if(pidComm[1] == 0) {
        close(fd[0]);
        dup2(fd[1], 2);                      /* LINE2 */
        dup2(fd[1], STDOUT_FILENO);         /* LINE3 */
        execv(command1, argv1);
        printf("%s: No such command\n", command1);
        exit(0);
    }
    else if(pidComm[1] > 0) {
        pidComm[0] = fork();

        if (pidComm[0] < 0) {
            printf("Can't create new process for pipeline\n");
            kill(pidComm[1], SIGKILL);
            return;
        }
        else if(pidComm[0] == 0) {
            close(fd[1]);
            dup2(fd[0], 0);                  /* LINE4 */
            dup2(fd[0], STDIN_FILENO);       /* LINE5 */
            execv(command2, argv2);
            printf("%s: No such command\n", command2);
            exit(0);
        }
        else if(pidComm[0] > 0) {
            close(fd[0]);
            close(fd[1]);
            wait(NULL);                      /* LINE6 */
            wait(NULL);                      /* LINE7 */
        }
    }
}
```

موفق باشید