Department of
Computer Engineering

Homework 1

Fall 2023

Dr. Javadi

Deadline: 1402/8/5

۱- به سوالات زیر پاسخ دهید.

الف) هدف از DMA چیست؟

ب) چگونه می توان سیستمی طراحی کرد که اجازه ی انتخاب یک سیستم عامل از چند سیستم عامل را هنگام بوت شدن به کاربر بدهد؟ برنامه ی bootstrap برای این منظور چه کاری باید انجام دهد؟

پ) توضیح دهید که تفاوت بین حالت کرنل و حالت کاربر چگونه به حفاظت و امنیت سیستم کمک میکند. کدام یک از دستورات زیر باید در حالت کرنل اجرا شوند؟

a. Set value of timer.

b. Read the clock.

c. Clear memory.

d. Issue a trap instruction.

e. Turn off interrupts.

f. Modify entries in device-status table.

g. Switch from user to kernel mode.

h. Access I/O device

د) در یک محیط multiprogramming و time-sharing چند کاربر به صورت همزمان سیستم را به اشتراک می گذارند و این وضعیت می تواند منجر به مشکلات امنیتی مختلف شود،2 مورد از این مشکلات را نام ببرید .

۲- به سوالات زیر پاسخ دهید.

الف) وقفه چیست؟ وقفه های سنکرون و آسنکرون را با هم مقایسه کنید .

ب) تفاوت های بین interrupt و trap را توضیح دهید .

پ) فرایند مدیریت یک وقفه از لحظه ایجاد شدن تا اتمام آن را توضیح دهید . فرض کنید وقفه متعدد نداریم و CPU مشغول انجام برنامه کاربر است.

۳- در هر یک از موارد زیر، پردازه از چه حالتی به چه حالت دیگری تغییر وضعیت می دهد؟(منظور از حالت، وضعیت های مختلف پردازه شامل running ,new, terminated ,ready ,waiting است)

الف) در حین اجرای پردازه، کاربر کلیدی را فشار داده و وقفه ای با اولویت بالا در سیستم اتفاق اتفاق می افتد.

ب) پردازه در حین اجرای کد خود، به جایی می رسد که نیاز به دریافت داده ها از طریق شبکه دارد.

پ) رویداد مربوط به یکی از دستگاههای ورودی/خروجی به پایان رسیده و داده های مورد نیاز پردازه حاضر می شوند.

ت) برنامه ای به زبان C، تابع exit را فراخوانی می کند.

ث) زمانبند سیستم عامل، پردازه ای را از صف انتظار خارج کرده و به آن اجازه اجرا می دهد.

٤- خروجی قطعه کدهای داده شده چیست؟ درخت پردازه های هر برنامه را رسم و راه حل خود را توضیح دهید؟

i
```
int main() {
    int i;
    for (i=0; i<3; ++i) {
        fork();
    }
    printf("Hello\n");
    return 0;
}
```

ii
```
int main() {
    if (fork() && fork()) {
        fork();
        fork();
    }
    printf("A");
    return 0;
}
```

iii
```
int main() {
    int i = 2;
    while (i > 0) {
        i--;
        if(fork()) {
            fork();
            printf("B");
        }
    }
return0;
}
```

iv
```
int main() {
    if (fork() || fork() && fork()) {
        fork();
        printf("C");
    }
    return 0;
}
```

٥-به سوالات زیر پاسخ کامل دهید.

الف) قطعه کد روبرو را در نظر بگیرید :

```
int main() {
    int count = 0;
    pid_t ret = fork;
    if (ret == 0) {
        printf(“count in child=%d\n”, count);
    }
    else {
        count = 1;
    }
    return 0;
}
```

اگر والد دستور "count = 1"را قبل از اجرای فرزند برای اولین بار اجرا می کند، مشخص کنید مقدار چاپ شده توسط کد بالا چقدر است؟ توضیح دهید.

ب) با توجه به پردازه های zombie و orphan به سوالات زیر پاسخ دهید :

- توضیح دهید که هر کدام از این پردازه ها چگونه ایجاد می شوند و تفاوت آنها با یکدیگر چیست؟
- با توجه به قطعه کد های زیر برای هر مورد مشخص کنید که بین orphan و zombie چه پردازه ای ایجاد می شود و جواب خود را به طور کامل توضیح دهید :

```
int main() {
    pid_t pid = fork;
    if (pid == 0) {
        printf(“child process with pid %d”, getpid());
        exit(0);
    }
    else {
        printf(“parent process with pid %d”, getpid());
        sleep(60);
    }
    return 0;
}
```

i

```
int main() {
    pid_t pid = fork;
    if (pid > 0) {
        printf(“parent process”);
    }
    else {
        sleep(60);
        printf(“child process”);
    }
    return 0;
}
```

ii

٦-در این تمرین قصد داریم تا پردازه های سیستم عامل لینوکس را مورد بررسی قرار بدهیم.

اولین دستوری که میخواهیم بررسی کنیم دستور"ps" است. این دستور اطلاعات پردازه ها را نمایش میدهد.

برای درک وضعیت فعلی فرآیندهای در حال اجرا سیستم خود دستور"aux ps" بیشترین مقدار اطلاعاتی که یک کاربر معمولاً نیاز دارد را نمایش می دهد.

این دستور را اجرا کنید و خروجی اش را مشاهده کنید.

در خروجی این دستور وضعیت هر پردازه(STAT) ، شماره هر پردازه(PID) و سایر اطلاعات موجود است.

شماره پردازه ها در خروجی دستور" ps" از 1 شروع شده است و می دانیم که یک پردازه جدید در سیستم عامل با استفاده از دستورات fork و exec ساخته می شود. بنابراین هر پردازه ای یک پردازه والد دارد.

برای یافتن شماره پردازه والد، میتوان از دستور" [pid] ps -f" استفاده کرد. این دستور PPIDکه شماره پردازه والد هست را نیز نمایش میدهد. از این دستور استفاده کنید و پردازه والد پردازه 1 را پیدا کنید.

دستور "[pid] pgrep -P" همه ی فرزندان یک پردازه را نمایش میدهد. از این دستور استفاده کنید و همه ی پردازه هایی که پردازه پدر مشترکی با پردازه 1 دارند را پیدا کنید.

همچنین با استفاده از دستور "pstree" میتوان ارتباط بین پردازه ها را نمایش داد.

تمامی مراحل و دستورات خواسته شده را انجام دهید و گزارش آن ارسال کنید. (امتیازی)

نکات مهم:

*مهلت ارسال تمرین ساعت 23:59 روز 1402/8/5 می باشد،با توجه به مهلت تجمیعی هفت روز تاخیر مجاز برای تمارین،امکان تمدید تمرین وجود ندارد،بنابراین توصیه می شود ارسال پاسخ های خود را به ساعات پایانی موکول نکنید.

*در صورت کشف هر گونه تقلب بار اول برای هر دو طرف نمره صفر لحاظ می شود و از دفعات بعد مشمول جریمه نیز می گردند.

*پاسخ های خود را در قالب یک فایل pdf یا zip با فرمت OS_HW1_StudentNumber.pdf یا OS_HW1_StudentNumber.zip ارسال نمایید،مانند:

OS_HW1_9931005.pdf

*در تمام سوالات پیاده سازی توابع fork و ... را مطابق با مطالب آموزش داده شده در کلاس در نظر بگیرید و اجرای آنها را موفقیت آمیز و بدون خطا لحاظ کنید.

*هر گونه سوال خود را می توانید در گروه پرسش و پاسخ درس از ما بپرسید.

Useful links for study:

https://www.geeksforgeeks.org/fork-practice-questions
https://ubuntu.com/tutorials/command-line-for-beginners
https://www.digitalocean.com/community/tutorials/process-management-in-linux