

به نام خدا

مسیح تنورساز 40006133

تکلیف اول (سیستم های عامل)

سوال 1 عملی / تئوری:

الف) مقایسه int, call

Int :

- مخفف interrupt است.
- این دستور برای وارد کردن یک نقطه ی وقفه (interrupt) به حالت کرنل (kernel mode) که سطح دسترسی بیشتری دارد، استفاده می شود.
- در این دستور نیاز هست تا رجیستر های سی پی یو را ذخیره کنیم.
- در این دستور مود تغییر کرده و وارد مود کرنل میشویم.

Call :

- این دستور در زمان اجرای برنامه برای فراخوانی تابع ها به کار می رود.
- با استفاده از دستور call ، برنامه از حالت اجرای کد خود به حالت اجرای تابع فراخوانی شده منتقل می شود.
- این دستور مود را تغییر نمی دهد و در مود یوزر استفاده میشود.

ب) مقایسه Ret, Iret

Iret :

- این دستور برای بازگشت از یک نقطه‌ی وقفه به برنامه اجرایی استفاده می‌شود.
- نیاز هست تا رجیسترهای پردازنده که موقع صدا زدن int ذخیره کردیم را بازگردانیم.
- این دستور معمولاً در پایان روند اجرای یک نقطه ی وقفه استفاده می‌شود.
- Iret اغلب بعد از اجرای دستور Int می‌آید و برعکس عملیات های Int را انجام میدهد.

Ret :

- این دستور برای بازگشت از یک تابع به برنامه اصلی یا تابع فراخواننده استفاده می‌شود.
- ret اغلب بعد از فراخوانی تابع با دستور call می‌آید و به خط بعد از اجرای آن تابع می‌رود.

ج) مدیریت استک در فراخوانی سیستم و فراخوانی تابع:

1. فراخوانی سیستم (System Call):

- مدیریت استک:
- هنگامی که یک فراخوانی سیستم انجام می‌شود، نیاز به تغییر حالت اجرای برنامه از حالت کاربری به حالت کرنل (kernel mode) داریم.
- مقادیر مربوط به وضعیت استک (مانند مقدار فعلی شماره آدرس برنامه) به کرنل استک منتقل می‌شوند.
- عملیات مدیریت استک در اینجا توسط سخت‌افزار (مانند واحد مدیریت حافظه) انجام می‌شود.
- مقادیر اولیه:
- شماره ی نقطه ی وقفه (interrupt number) مورد نظر برای فراخوانی سیستم به عنوان یک پارامتر به تابع سیستم منتقل می‌شود.
- مقدارهای دیگر نیاز به مدیریت استک در حالت کرنلی دارند.

2. فراخوانی تابع (Procedure Call):

- مدیریت استک:

- هنگامی که یک تابع فراخوانی می‌شود، آدرس بازگشت و متغیرهای محلی تابع بر روی یوزر استک ذخیره می‌شوند.

- عملیات مدیریت استک در اینجا توسط کد مربوط به تابع و زبان برنامه‌نویسی انجام می‌شود.

- مقادیر اولیه:

- مقدار بازگشت (return address) که نشان‌دهنده آدرسی است که برنامه باید به آن بازگردد پس از اتمام تابع، بر روی استک قرار می‌گیرد.

- متغیرهای محلی تابع با استفاده از استک ذخیره می‌شوند.

- مقادیر دیگر متغیرهای پارامتر ورودی نیز ممکن است بر روی استک قرار گیرند.

در هر دو مورد، مدیریت استک نیازمند پشتیبانی سخت‌افزار و نرم‌افزار است. در فراخوانی تابع، کد مربوط به تابع و زبان برنامه‌نویسی این مسئولیت را به عهده دارد (نرم‌افزاری)، در حالی که در فراخوانی سیستم، این کار به صورت اصلی توسط سخت‌افزار و واحد مدیریت حافظه انجام می‌شود.

(د)

```
cd "/home/masih/Desktop/oslab/" && gcc test.c -o test && "/home/masih/Desktop/oslab/test"
❶ masih@masih-fun-life:~/Desktop/oslab$ cd "/home/masih/Desktop/oslab/" && gcc test.c -o test && "/home/masih/Desktop/oslab/test"
test.c:2:10: fatal error: time.h : No such file or directory
  2 | #include <time.h>
    |          ^~~~~~
compilation terminated.
❷ masih@masih-fun-life:~/Desktop/oslab$ cd "/home/masih/Desktop/oslab/05_40006133_04/" && gcc test.c -o test && "/home/masih/Desktop/oslab/05_40006133_04/test"
Procedure call (10000 times): 0.000016 sec
System call (10000 times): 0.005145 sec
❸ masih@masih-fun-life:~/Desktop/oslab/05_40006133_04/q1$
```

همانطور که قابل مشاهده است در سیستم کال ما زمان بیشتری را صرف کرده ایم. دلیلش این است که برای سیستم کال ما تغییر مود داریم و این overhead باعث میشود تا زمان بیشتری نسبت به procedure call مصرف شود.