

تمرین اول سیستم‌های عامل

اشکان شکيبا (9931030)

سوال اول

الف) به فرایندی که در آن، داده‌ها بدون گذر از پردازنده، مستقیماً بین سایر دستگاه‌ها (معمولاً حافظه و دستگاه‌های ورودی و خروجی) جابجا می‌شوند، Direct Memory Access یا به اختصار DMA گفته می‌شود.

ب) با درگیر نکردن پردازنده برای انتقال داده‌ها بین IO و حافظه و جابجایی مستقیم آنها از طریق DMA، به طوری که نیازی به پردازش دستورات مربوط به لود داده‌ها در رجیستر و سپس استور آنها در مقصد نیست.

پ) بله، ممکن است هر دو برنامه نیاز به دسترسی به منابع یکسانی داشته باشند که در صورت عدم مدیریت می‌تواند منجر به بروز مشکل شود. برای جلوگیری از این موضوع می‌توان از قفل‌های سخت‌افزاری و نرم‌افزاری به منظور جلوگیری از دسترسی همزمان به منابع استفاده کرد.

د) در این روش، با اتصال یک دستگاه با دسترسی DMA به سیستم (که می‌تواند یک USB Flash، کارت شبکه، کارت صدا و یا موارد مختلف دیگر باشد)، دسترسی مستقیم دستگاه مورد استفاده برای حمله به حافظه فراهم شده و می‌تواند با دسترسی به اطلاعات حساس حافظه، اقدام به سرقت

داده‌های حساس کند و یا با حذف یا ویرایش داده‌های مورد نیاز سیستم، اختلال پدید بیاورد.

برای جلوگیری از DMA Attack می‌توان از راهکارهایی مانند قفل کردن منابع، رمزنگاری داده‌ها و اعمال محدودیت دسترسی به حافظه برای دستگاه‌هایی که از DMA استفاده می‌کنند، بهره برد.

سوال دوم

الف) در خط ۱۴، استرینگ message در خروجی استاندارد یا به عبارتی ترمینال، نوشته می‌شود. (عملیات write)

در خط ۱۸، برنامه با کد وضعیت صفر، به پایان می‌رسد. (عملیات exit 0)

ب) عدد 0x4 مربوط به write و عدد 0x1 مربوط به exit است. بنابراین این قطعه کد استرینگ message را در خروجی استاندارد نمایش داده و سپس با کد صفر خاتمه می‌یابد.

پ) برای انتقال پارامترها از روش معمول در سیستم‌های x86 استفاده شده است که در آن پارامترها از طریق رجیسترهای خاصی که به آنها اختصاص داده شده است، به سرویس منتقل می‌شوند.

در بخش اول رجیستر eax دربرگیرنده کد سرویس، رجیستر ebx شامل فایل دسکریپتور، رجیستر ecx استرینگ مورد نظر و رجیستر edx طول استرینگ را ذخیره می‌کند.

در بخش دوم نیز رجیستر eax دربرگیرنده کد سرویس و رجیستر ebx مربوط به کد وضعیت است.

د) kernel mode، چرا که برای انجام هر دو عملیات write و exit نیاز به دسترسی‌هایی وجود دارد که در user mode به دلیل محدودیت‌های آن، فراهم نیست.

سوال سوم

دو پردازنده‌های دارای بیش از دو حالت عملکرد:

۱) پردازنده ARM Cortex-A:

این پردازنده دارای پنج حالت عملکرد مختلف است:

User mode: برای اجرای برنامه‌های کاربری.

System mode: برای اجرای سیستمی برنامه‌ها که دسترسی به تمام سخت‌افزارهای سیستم دارند.

FIQ mode: برای پردازش سریع آماده‌سازی درخواست‌های تعریف شده

IRQ mode: برای پردازش آماده‌سازی درخواست‌های معمولی

Supervisor mode: برای اجرای سیستمی برنامه‌ها که دسترسی به سخت‌افزارهای خاص دارند.

۲) پردازنده Intel x86:

این پردازنده دارای دو حالت عملکرد جدید است:

Real mode: حالتی که برای سازگاری با پردازنده‌های قدیمی‌تر و مواردی مانند بوت کردن سیستم استفاده می‌شود.

Protected mode: حالتی که برای اجرای سیستم عامل و برنامه‌های محدود به سطح کاربری استفاده می‌شود. این حالت اجازه می‌دهد که سیستم عامل دسترسی به حافظه و تمامی سخت‌افزارهای سیستم داشته باشد و قابلیت ایجاد چندین حالت دیگر نیز دارد.

استفاده از حالت‌های مختلف پردازنده‌ها اجازه می‌دهد که برنامه‌های مختلف در سطوح مختلف دسترسی به سیستم داشته باشند و همچنین بهینه‌سازی عملکرد سیستم را تسهیل می‌کند.

سوال چهارم

الف) ابتدا firmware برنامه bootloader را لود کرده که آن هم وظیفه لود و راه‌اندازی kernel را دارد. Kernel نیز پس از شروع به کار، برنامه init را اجرا می‌کند تا سایر برنامه‌های مورد نیاز را اجرا کند. برخی از وظایف این برنامه بارگذاری سیستم عامل، انتقال کنترل به kernel و اعمال برخی تنظیمات سخت‌افزاری است.

ب) برای این منظور، ابتدا یک برنامه bootloader کوچک و ساده لود و اجرا می‌شود که پیش‌تر در کانفیگ آن، kernel ها و chainloader های مورد نظر معرفی شده‌اند و با نمایش منویی به کاربر امکان لود گزینه‌های مختلف با پارامترهایی که از پیش مشخص شده‌اند و قابل ویرایش نیز هستند، می‌دهد.

سوال پنجم

الف) در multiprocessing، چندین برنامه به صورت همزمان در حافظه قرار می‌گیرند و در هر لحظه، یک برنامه اجرا می‌شود. اگر برنامه اول در حال انجام عملیات io باشد، برنامه دوم می‌تواند در انتظار باشد و از پردازنده درخواست خود را انجام دهد. با این روش، کارایی سیستم بالاتر می‌رود.

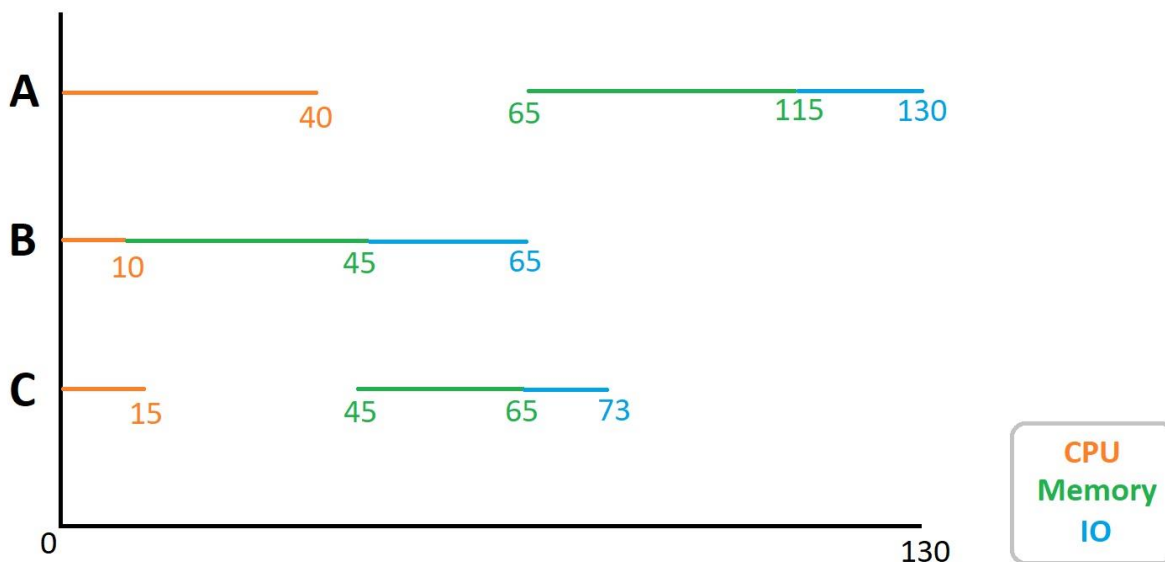
در multitasking، نیازی به منتظر ماندن برنامه‌ها نیست، بلکه چندین برنامه به صورت همزمان و به صورت موازی اجرا می‌شوند. در این روش، پردازنده برای مدتی به هر برنامه تخصیص داده می‌شود و سپس به برنامه بعدی می‌رود. این فرایند به صورت پیوسته تکرار می‌شود تا همه برنامه‌ها به انتهای اجرای خود برسند.

در کل، multiprocessing به معنای اجرای همزمان چندین برنامه در یک سیستم عامل است و multitasking به معنای اجرای همزمان و همروند برنامه‌ها بر روی یک پردازنده است.

ب) در یک سیستم با یک پردازنده و یک هسته که از multiprocessing پشتیبانی می‌کند، در هر لحظه تنها یک پردازنده می‌تواند در حالت running باشد. با این حال، با استفاده از روش‌هایی مانند time-sharing و context switching، سیستم عامل می‌تواند به نظر برسد که چندین پردازنده همزمان در حال اجرا هستند و کارایی بالاتری داشته باشد. در واقع، سیستم عامل به ترتیب و به طور مداوم، چندین پردازنده را به صورت موازی اجرا می‌کند و بین آن‌ها جابجا می‌شود. به این روش multitasking یا time-sharing نیز گفته می‌شود.

ج) با توجه به صورت سوال «یک سیستم چند پردازنده و دو هسته‌ای» داریم. یعنی بیش از یک پردازنده که هر کدام دو هسته دارند، بنابراین

حداقل چهار هسته داریم؛ پس در ابتدا هر کدام از سه پردازش بر روی یک هسته شروع به کار می‌کنند که فرض می‌کنیم A بر روی هسته اول از پردازنده اول، B بر روی هسته دوم از پردازنده اول و C بر روی هسته اول از پردازنده دوم اجرا شوند. پس از ۱۰ نانوثانیه پردازش پردازش B به پایان می‌رسد و به سراغ حافظه می‌رود و تا زمان ۴۵ نانوثانیه، حافظه درگیر عملیات مربوط به B است. تا آن زمان پردازش هر دو پردازش C و A به پایان رسیده است که با توجه به اتمام زودتر C عملیات حافظه مربوط به آن شروع شده و B نیز به سراغ عملیات IO می‌رود که تا زمان ۶۵ نانوثانیه طول می‌کشد. همچنین کار C با حافظه پس از ۲۰ ثانیه یعنی در همان زمان ۶۵ نانوثانیه تمام شده و حافظه را در اختیار A می‌گذارد و خود به سراغ IO می‌رود که تا زمان ۷۳ ثانیه ادامه می‌یابد. A نیز ۵۰ نانوثانیه با حافظه کار می‌کند و در زمان ۱۱۵ نانوثانیه به سراغ IO می‌رود و پس از ۱۵ نانوثانیه در زمان ۱۳۰ نانوثانیه اجرای آن به پایان می‌رسد. بنابراین ۱۳۰ نانوثانیه زمان لازم است تا هر سه پردازش به طور کامل اجرا شوند.



حال مقدار CPU Utilization را برای پردازنده اول و دوم محاسبه می‌کنیم.

پردازنده اول ۴۰ نانوثانیه مشغول به کار بوده که حدوداً ۲۹/۶ درصد از زمان کل اجراست.

پردازنده دوم نیز ۱۵ نانوثانیه مشغول به کار بوده که حدوداً ۱۱/۱ درصد از زمان کل اجراست.

همچنین مقدار CPU Utilization برای هسته‌ها بدین شکل است:

هسته اول از پردازنده اول ۲۹/۶ درصد

هسته دوم از پردازنده اول ۷/۴ درصد

هسته اول از پردازنده دوم ۱۱/۱ درصد

بخش عملی

```
ashkan@ubuntu: ~/OS-course/HW1$ wget https://raw.githubusercontent.com/Donders-Institute/hpc-wiki-v2/master/docs/linux/exercise/gcutError_recon_all.log
--2023-03-03 08:09:09-- https://raw.githubusercontent.com/Donders-Institute/hpc-wiki-v2/master/docs/linux/exercise/gcutError_recon_all.log
Resolving raw.githubusercontent.com (raw.githubusercontent.com)... 185.199.110.133, 185.199.108.133, 185.199.109.133, ...
Connecting to raw.githubusercontent.com (raw.githubusercontent.com)|185.199.110.133|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 469166 (458K) [text/plain]
Saving to: 'gcutError_recon-all.log'

gcutError_recon-all.log      19%[=====>] 87.16K 36.8KB/s
gcutError_recon-all.log    100%[=====>] 458.17K 42.4KB/s   in 11s

2023-03-03 08:09:21 (41.0 KB/s) - 'gcutError_recon-all.log' saved [469166/469166]

ashkan@ubuntu:~/OS-course/HW1$ cat gcutError_recon-all.log | grep Subject05
/home/language/dansha/Studies/LaminarWord/SubjectData/Subject05/FreeSurfer
-subjid FreeSurfer -i /home/language/dansha/Studies/LaminarWord/SubjectData/Subject05/Scans/Anatomical/MP2RAGE/MP2RAGE.nii -all
setenv SUBJECTS_DIR /home/language/dansha/Studies/LaminarWord/SubjectData/Subject05
/home/language/dansha/Studies/LaminarWord/SubjectData/Subject05/FreeSurfer
mri_convert /home/language/dansha/Studies/LaminarWord/SubjectData/Subject05/Scans/Anatomical/MP2RAGE/MP2RAGE.nii /home/language/dansha/Studies/LaminarWord/SubjectData/Subject05/FreeSurfer/mri/orig/001.mgz
mri_convert /home/language/dansha/Studies/LaminarWord/SubjectData/Subject05/Scans/Anatomical/MP2RAGE/MP2RAGE.nii /home/language/dansha/Studies/LaminarWord/SubjectData/Subject05/FreeSurfer/mri/orig/001.mgz
reading from /home/language/dansha/Studies/LaminarWord/SubjectData/Subject05/Scans/Anatomical/MP2RAGE/MP2RAGE.nii...
writing to /home/language/dansha/Studies/LaminarWord/SubjectData/Subject05/FreeSurfer/mri/orig/001.mgz...
/home/language/dansha/Studies/LaminarWord/SubjectData/Subject05/FreeSurfer/mri/orig/001.mgz
cp /home/language/dansha/Studies/LaminarWord/SubjectData/Subject05/FreeSurfer/mri/orig/001.mgz /home/language/dansha/Studies/LaminarWord/SubjectData/Subject05/FreeSurfer/mri/rawavg.mgz
/home/language/dansha/Studies/LaminarWord/SubjectData/Subject05/FreeSurfer
mri_convert /home/language/dansha/Studies/LaminarWord/SubjectData/Subject05/FreeSurfer/mri/rawavg.mgz /home/language/dansha/Studies/LaminarWord/SubjectData/Subject05/FreeSurfer/mri/orig.mgz --conform
```

بنابراین خط اول خطای مربوطه این است:

home/language/dansha/Studies/LaminarWord/SubjectData/Subject05/FreeSurfer/