OS

HW#1

Seyed Alireza Fatemi Jahromi

95102062

1

بله، می تواند وجود داشته باشد. مثلا سیستم عامل از مکانیزم polling استفاده کند که باعث تلف شدن وقت سیستم عامل می شود و یا اینکه هر پروسه برای اینکه اجرا شود باید منتظر بماند تا اولویت بالاتری نسبت به پروسه های دیگر داشته باشد.

منابع:

http://faculty.salina.k-state.edu/tim/ossg/Introduction/OSworking.html

http://www.cse.iitm.ac.in/~chester/courses/15o_os/slides/5_Interrupts.pdf

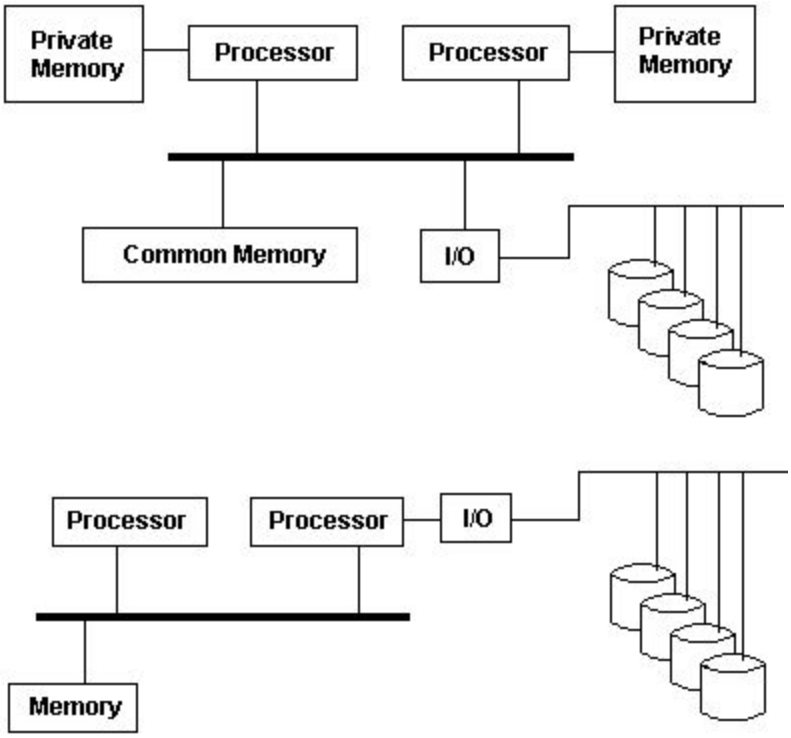
چند هستگی به یک پردازنده با چند هسته اشاره می کند. چند پردازندگی به یک سیستم با چند پردازنده اشاره می کند.

در حالت چند هستگی cache coherency بیشتر است و به صورت کلی انرژی کمتری مصرف می کند. اجرای یک برنامه در حالت چندهستگی سریعتر است اما اجرای چند برنامه در حالت چند پردازندگی سریع تر است.

در حالت چند پردازندگی reliability بیشتر است زیرا در صورت از کار افتادن یک پردازنده، پردازنده های دیگر وجود دارند. اگر فشار زیادی بر روی یک پردازنده بیفتد باعث افزایش مصرف باتری می شود در صورتی که اگر فشار بین چند پردازنده تقسیم شود مصرف باتری کمتر می شود. چونکه bus و memory و I/O devices مشترک هستند، throughput کاهش می یابد. برای استفاده بهینه باید main memory بالایی داشته باشیم. بعضی سیستم عامل ها ممکن است از چند پردازندگی پشتیبانی نکنند.

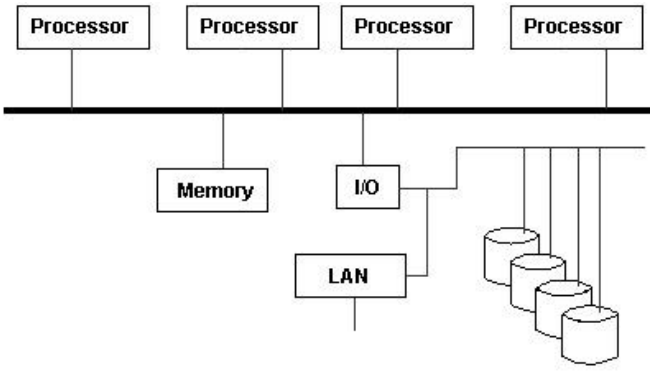
برای سیستم های خانگی چند هستگی مناسب تر است اما برای سیستم های پردازشی سنگین سیستم های چند پردازنده مناسب هستند.

چند پردازندگی نامتقارن حالتی هست که همه پردازنده ها مشابه یکدیگر کار نمی کنند. در این حالت یک پردازنده (master processor) وظیفه اجرای سیستم عامل را دارد. رابطه بین پردازنده ها master-slave است. معمولا در این حالت هر پردازنده فضای مموری خود را دارد.



اگر مثلا در کنار پردازنده اصلی یک پردازنده دیگر برای انجام کار های امنیتی و رمزگذاری استفاده شود، این معماری مناسب است.

چند پردازندگی متقارن:



SYMMETRIC MULTIPROCESSING
VERSUS
ASYMMETRIC MULTIPROCESSING

SYMMETRIC MULTIPROCESSING	ASYMMETRIC MULTIPROCESSING
Processing of programs by multiple processors that share a common operating system and memory	Processing of programs by multiple processors that function according to the master-slave relationship
All the processors are treated equally	Processors are not treated equally
Processors take processes from the ready queue - each processor can have separate ready queues	Master processor assigns processes to the slave processors
Processors communicate with each other by the shared memory	Processors communicate with the master processor
All processors have the same architecture	Architecture can be different for each processor
Not as easy to design or handle	Easier to design and handle
Comparatively costly	Cheaper

Visit www.PEDIAA.com

سیستم های مبتنی بر این معماری به راحتی scalable هستند. برای اکثر سیستم ها از همین معماری استفاده می شود زیرا مانند معماری متقارن نیاز به یک طراحی خاص ندارد.

منابع:

http://ohlandl.ipv7.net/CPU/ASMP_SMP.html

https://pediaa.com/what-is-the-difference-between-symmetric-and-asymmetric-multiprocessing/

https://blog.qburst.com/2013/04/symmetric-vs-asymmetric-multiprocessing/

https://afteracademy.com/blog/what-is-the-difference-between-a-multicore-system-and-a-multiprocessor-system

3

دستورات اسمبلی زیر (Privileged Level Instructions) در حالت کرنل فقط اجرا می شوند:

Instruction	**Description**
LGDT	Loads an address of a GDT into GDTR
LLDT	Loads an address of a LDT into LDTR
LTR	Loads a Task Register into TR
MOV *Control Register*	Copy data and store in Control Registers
LMSW	Load a new Machine Status WORD
CLTS	Clear Task Switch Flag in Control Register CR0
MOV *Debug Register*	Copy data and store in debug registers
INVD	Invalidate Cache without writeback
INVLPG	Invalidate TLB Entry
WBINVD	Invalidate Cache with writeback
HLT	Halt Processor
RDMSR	Read Model Specific Registers (MSR)
WRMSR	Write Model Specific Registers (MSR)
RDPMC	Read Performance Monitoring Counter
RDTSC	Read time Stamp Counter

HLT - Halt

پردازش دستورات توسط پردازنده را متوقف می سازد. با ریستارت شدن یا non-maskable interrupt پردازنده به پردازش ادامه می دهد.

INVD - Invalidate Cache without writeback

کش های پردازنده را invalidate می کند و داده های داخل کش دیگر بر روی دیسک نوشته نمی شوند. از این دستور زمانی استفاده می شود که کش های پردازنده به عنوان یک مموری موقت استفاده می شود ((Cache-as-RAM (CAR)) و باید invalidate شود به جای اینکه بر روی مموری نوشته شود.

WBINVD - Write Back and Invalidate Cache

قسمت های تغییر یافته کش را بر روی مموری می نویسد و کش را invalidate می کند.

منابع:

http://www.brokenthorn.com/Resources/OSDev23.html

http://faydoc.tripod.com/cpu/hlt.htm

https://pdos.csail.mit.edu/6.828/2010/readings/i386/HLT.htm

https://stackoverflow.com/questions/41775371/what-use-is-the-invd-instruction

https://www.felixcloutier.com/x86/invd

https://www.felixcloutier.com/x86/wbinvd

https://blog.codinghorror.com/understanding-user-and-kernel-mode/

تمرین عملی

دستورات مراحل مختلف در makefile قرار دارند.

```
obj-m+=first_module.o

compile:
	make -C /lib/modules/$(shell uname -r)/build/ M=$(PWD)
modules
install:
	make -C /lib/modules/$(shell uname -r)/build/ M=$(PWD)
modules_install
add:
	sudo insmod first_module.ko
	sudo depmod
	sudo modprobe first_module
remove:
	sudo rmmod first_module
	sudo modprobe -r first_module
	sudo rm -f /etc/modules-load.d/reboot.conf
	sudo rm -f /lib/modules/$(shell uname
-r)/kernel/drivers/pci/pcie/first_module.ko
test:
	sudo dmesg -C
	sudo insmod first_module.ko
	sudo rmmod first_module.ko
	dmesg
clean:
	make -C /lib/modules/$(shell uname -r)/build/ M=$(PWD)
clean
	rm -rf   Module.symvers modules.order
```

کامپایل ماژول و نصب و تست

```
root@debian:~/os-hw1# make compile
make -C /lib/modules/3.16.84/build/ M=/root/os-hw1 modules
make[1]: Entering directory '/root/hw1/linux-3.16.84'
  CC [M]  /root/os-hw1/first_module.o
/root/os-hw1/first_module.c: In function ‘init’:
/root/os-hw1/first_module.c:28:2: warning: format ‘%d’ expects argument of type ‘int’, but argument 2 has type ‘long int’ [-Wfo
rmat=]
  printk(KERN_INFO "Date: %d/%d/%d\n", time_tm.tm_year + 1900, time_tm.tm_mon + 1, time_tm.tm_mday);
  ^
  Building modules, stage 2.
  MODPOST 1 modules
  CC      /root/os-hw1/first_module.mod.o
  LD [M]  /root/os-hw1/first_module.ko
make[1]: Leaving directory '/root/hw1/linux-3.16.84'
root@debian:~/os-hw1# make install
make -C /lib/modules/3.16.84/build/ M=/root/os-hw1 modules_install
make[1]: Entering directory '/root/hw1/linux-3.16.84'
  INSTALL /root/os-hw1/first_module.ko
  DEPMOD  3.16.84
make[1]: Leaving directory '/root/hw1/linux-3.16.84'
root@debian:~/os-hw1# make test
sudo dmesg -C
sudo insmod first_module.ko
sudo rmmod first_module.ko
dmesg
[  366.199832] Time:  16:48:59
[  366.199834] Date: 2020/10/14
[  366.204057] Goodbye!
root@debian:~/os-hw1# make add
sudo insmod first_module.ko
sudo depmod
sudo modprobe first_module
root@debian:~/os-hw1# lsmod | grep first
first_module           12357  0
root@debian:~/os-hw1#
```

تست ماژول و اضافه کردن ماژول و مشاهده اطلاعات و حذف ماژول

```
root@debian:~/os-hw1# make add
sudo insmod first_module.ko
sudo depmod
sudo modprobe first_module
root@debian:~/os-hw1# lsmod | grep first
first_module           12357  0
root@debian:~/os-hw1# modinfo first_module
filename:       /lib/modules/3.16.84/extra/first_module.ko
version:        0.1
description:    Prints system's date and time
author:         Seyed Alireza Fatemi Jahromi
license:        MIT
srcversion:     E55F6F09ADD706444A50F9B
depends:
retpoline:      Y
vermagic:       3.16.84 SMP mod_unload modversions 686
root@debian:~/os-hw1# make remove
sudo rmmod first_module
sudo modprobe -r first_module
sudo rm -f /etc/modules-load.d/reboot.conf
sudo rm -f /lib/modules/3.16.84/kernel/drivers/pci/pcie/first_module.ko
root@debian:~/os-hw1# lsmod | grep first
root@debian:~/os-hw1#
```

تابع init تابعی است که موقع اضافه شدن ماژول به کرنل فراخوانی می شود. تابع exit است تابعی است که موقع حذف شدن ماژول از کرنل فراخوانی می شود.

در این دو خط نیز این دو تابع را register می کنیم.

```
module_init(init);
module_exit(exit);
```

تابع printk شبیه تابع printf می باشد. در تابع printk می توانیم log level تعریف کنیم. تمامی پیام های printk در بافر لاگ کرنل نوشته می شوند که در اینجا /dev/kmsg نوشته export می شود و با دستور dmesg آن را می توانیم بخوانیم.

یکی از مشکلات مواجهه ssl error بود که با مهاجرت به سیستم debian 8.1.0 که کرنل آن ورژن پایین تری نسبت به ubuntu server 16.04.6 داشت، رفع شد. مثل اینکه در ورژن های جدید کرنل برای اضافه کردن ماژول به کرنل نیاز است که ماژول sign شده باشد.

یکی دیگر از مشکلات مواجهه با ارور زیر بعد از دستور modprobe بود:

modprobe: FATAL: Module first_module not found in directory …

که با اجرای دستور sudo depmod بعد از insmod حل شد. این دستور module dependency list را دوباره می سازد.

منابع:

https://www.geeksforgeeks.org/linux-kernel-module-programming-hello-world-program/

https://stackoverflow.com/questions/26069352/custom-linux-kernel-module-to-display-year-date-and-time

https://github.com/umlaeute/v4l2loopback/issues/139

https://blog.sourcerer.io/writing-a-simple-linux-kernel-module-d9dc3762c234

https://stackoverflow.com/questions/55566038/how-can-i-print-current-time-in-kernel

https://stackoverflow.com/questions/5077192/how-to-get-current-hour-time-of-day-in-linux-kernel-space

https://docs.huihoo.com/doxygen/linux/kernel/3.7/timeconv_8c.html

https://www.kernel.org/doc/html/latest/core-api/timekeeping.html#c.ktime_get_ts64

https://www.kernel.org/doc/html/latest/core-api/printk-basics.html

https://stackoverflow.com/questions/34800731/module-not-found-when-i-do-a-modprobe