操作系统原理 September 1, 2011

Homework 1

计算机 92 班 戴唯思 09055029

1.5

> Describe the differences between symmetric and asymmetric multiprocessing. What are three advantages and one disadvantage of multiprocessor systems?

Table 1: Comparing SMP with ASMP

Symmetric Multiprocessing(SMP)	**Asymmetric Multiprocessing(ASMP)**
CPUs are all equal	One master with slaves
Shared memory space	Divided segments of memory space
A single main memory	Private memory and common memory
Easy to implement	Difficult to implement
Doing paralleled job	Requires specialized knowledge on tasks
	unavailable in a general purpose OS, e.g. Linux

The advantages:

1. Better performance and fault-talerance on calculating,
2. Capable of running paralleled tasks,
3. Less expense on supporting hardware.

The disadvantage:

1. Additional time spent on tuning programs to run on.

1.10

> What is the purpose of interrupts? What are the differences between a trap and an interrupt? Can traps be generated intentionally by a user program? If so, for what purpose?

When reading inputs from external devices, polling loops consume a lot of CPU time if there were not hardware interrupts.

A *trap* is an exception in a user process, while an *interrupt* is often generated by hardware.

Traps are usually caused by illegal operations such as “divided by zero” or “segmentation fault”, or requesting a kernel routine running on a higher priority. A trap in the latter case are named a *system call.*

Traps can be called in userspace and may result in switches to the kernel, but only in kernel space happen interrupts .

Traps are handled synchronously[1], but interrupts are handled asynchronously[2].

As said above, traps can be called intentionally to execute a specific kernel routine.

[1] user code gets suspended when the traps are being handled

[2] but not that absolutely

1.17

Define the essential properties of the following types of operating systems:

a. Batch

b. Interactive

c. Time sharing

d. Real time

e. Network

a. **Batch**
Jobs are similar and are off-line[3], and should be automatically sequenced. When enough tasks are in the queue, CPU and I/O are kept busy in order to optimize in-all performance. Jobs can be useful when little interaction is needed in the jobs.

b. **Interactive**
Transactions are short and can end in a short time, while the operator submits and waits for the result.

c. **Time sharing**
Supports multi-tasking, and CPU resources are scheduled for different tasks. Policies may differ for particular types of jobs. The OS controls the CPU to switch between users/tasks.

d. **Real time**
Supports a set of particular jobs, and can handle them very well, e.g., returns the result within a predictable time, which is usually very short.

e. **Network**
Supports network features, e.g., file and devices sharing.

References

1. http://web.engr.oregonstate.edu/~pancake/cs411/hw1sol.s99.html

2. http://tldp.org/LDP/lkmpg/2.4/html/c1294.htm

3. https://secure.wikimedia.org/wikipedia/en/wiki/Asymmetric_multiprocessing

4. https://secure.wikimedia.org/wikipedia/en/wiki/Symmetric_multiprocessing

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

6. http://www.quora.com/Does-software-need-to-be-designed-to-run-on-multiple-processors-to-be-efficient

7. https://secure.wikimedia.org/wikipedia/en/wiki/Interrupt

8. http://stackoverflow.com/questions/3149175/what-is-the-diffrence-between-trap-and-interrupt

9. http://stackoverflow.com/questions/3425085/the-difference-between-call-gate-interrupt-gate-trap-gate

[3] offline problems does not require a solution as soon as they are raised

10. https://secure.wikimedia.org/wikipedia/en/wiki/Kernel_trap

11. http://en.allexperts.com/q/Unix-Linux-OS-1064/2009/3/Interrupts-Traps.htm

12. http://www.linuxjournal.com/article/3985

13. http://lxr.free-electrons.com/source/arch/x86/kernel/traps.c

14. http://stackoverflow.com/questions/4821286/software-traps-vs-hardware-traps

15. http://shrutikohli.com/pdf/os.pdf

16. http://faculty.ksu.edu.sa/mougy/Documents/CSC227-HW2-Sol.pdf