

Multiprocessing

From Wikipedia, the free encyclopedia

Multiprocessing is the use of two or more central processing units (CPUs) within a single computer system.^{[1][2]} The term also refers to the ability of a system to support more than one processor or the ability to allocate tasks between them.^[3] There are many variations on this basic theme, and the definition of multiprocessing can vary with context, mostly as a function of how CPUs are defined (multiple cores on one die, multiple dies in one package, multiple packages in one system unit, etc.).

According to some on-line dictionaries, a **multiprocessor** is a computer system having two or more processing units (multiple processors) each sharing main memory and peripherals, in order to simultaneously process programs.^{[4][5]} A 2009 textbook defined multiprocessor system similarly, but noting that the processors may share "some or all of the system's memory and I/O facilities"; it also gave **tightly coupled system** as a synonymous term.^[6]

At the operating system level, *multiprocessing* is sometimes used to refer to the execution of multiple concurrent processes in a system, with each process running on a separate CPU or core, as opposed to a single process at any one instant.^{[7][8]} When used with this definition, multiprocessing is sometimes contrasted with multitasking, which may use just a single processor but switch it in time slices between tasks (i.e. a time-sharing system). Multiprocessing however means true parallel execution of multiple processes using more than one processor.^[8] Multiprocessing doesn't necessarily mean that a single process or task uses more than one processor simultaneously; the term parallel processing is generally used to denote that scenario.^[7] Other authors prefer to refer to the operating system techniques as multiprogramming and reserve the term *multiprocessing* for the hardware aspect of having more than one processor.^{[2][9]} The remainder of this article discusses multiprocessing only in this hardware sense.

In Flynn's taxonomy, multiprocessors as defined above are MIMD machines.^{[10][11]} As they are normally construed to be tightly coupled (share memory), multiprocessors are not the entire class of MIMD machines, which also contains message passing multicomputer systems.^[10]



Contents

- 1 Pre-history
- 2 Key topics
 - 2.1 Processor symmetry
 - 2.2 Instruction and data streams
 - 2.3 Processor coupling
 - 2.3.1 Tightly coupled multiprocessor system
 - 2.3.2 Loosely coupled multiprocessor system
- 3 See also
- 4 References

Pre-history

According to a 1985 article in *Byte*, possibly the first expression of the idea of multiprocessing is found in the 1842 words of Luigi Federico Menabrea, which said about Charles Babbage's analytical engine (as translated by Ada Lovelace): "the machine can be brought into play so as to give several results at the same time, which will greatly abridge the whole amount of the processes."^[12]

Key topics

Processor symmetry

In a **multiprocessing** system, all CPUs may be equal, or some may be reserved for special purposes. A combination of hardware and operating system software design considerations determine the symmetry (or lack thereof) in a given system. For example, hardware or software considerations may require that only one particular CPU respond to all hardware interrupts, whereas all other work in the system may be distributed equally among CPUs; or execution of kernel-mode code may be restricted to only one particular CPU, whereas user-mode code may be executed in any combination of processors. Multiprocessing systems are often easier to design if such restrictions are imposed, but they tend to be less efficient than systems in which all CPUs are utilized.

Systems that treat all CPUs equally are called symmetric multiprocessing (SMP) systems. In systems where all CPUs are not equal, system resources may be divided in a number of ways, including asymmetric multiprocessing (ASMP), non-uniform memory access (NUMA) multiprocessing, and clustered multiprocessing.

Instruction and data streams

In multiprocessing, the processors can be used to execute a single sequence of instructions in multiple contexts (single-instruction, multiple-data or SIMD, often used in vector processing), multiple sequences of instructions in a single context (multiple-instruction, single-data or MISD, used for redundancy in fail-safe systems and sometimes applied to describe pipelined processors or hyper-threading), or multiple sequences of instructions in multiple contexts (multiple-instruction, multiple-data or MIMD).

Processor coupling

Tightly coupled multiprocessor system

Tightly coupled multiprocessor systems contain multiple CPUs that are connected at the bus level. These CPUs may have access to a central shared memory (SMP or UMA), or may participate in a memory hierarchy with both local and shared memory (SM)(NUMA). The IBM p690 Regatta is an example of a high end SMP system. Intel Xeon processors dominated the multiprocessor market for business PCs and were the only major x86 option until the release of AMD's Opteron range of processors in 2004. Both ranges of processors had their own onboard cache but provided access to shared memory; the Xeon processors via a common pipe and the Opteron processors via independent pathways to the system RAM.

Chip multiprocessors, also known as multi-core computing, involves more than one processor placed on a single chip and can be thought of the most extreme form of tightly coupled multiprocessing. Mainframe systems with multiple processors are often tightly coupled.

Loosely coupled multiprocessor system

Loosely coupled multiprocessor systems (often referred to as clusters) are based on multiple standalone single or dual processor commodity computers interconnected via a high speed communication system (Gigabit Ethernet is common). A Linux Beowulf cluster is an example of a loosely coupled system.

Tightly coupled systems perform better and are physically smaller than loosely coupled systems, but have historically required greater initial investments and may depreciate rapidly; nodes in a loosely coupled system are usually inexpensive commodity computers and can be recycled as independent machines upon retirement from the cluster.

Power consumption is also a consideration. Tightly coupled systems tend to be much more energy efficient than clusters. This is because considerable economy can be realized by designing components to work together from the beginning in tightly coupled systems, whereas loosely coupled systems use components that were not necessarily intended specifically for use in such systems.

Loosely coupled systems have the ability to run different operating systems or OS versions on different systems.

See also

- Symmetric multiprocessing
- Asymmetric multiprocessing
- Multi-core processor
- BMDFM – Binary Modular Dataflow Machine, a SMP MIMD runtime environment
- Software lockout
- OpenHMPP

References

1. Raj Rajagopal (1999). *Introduction to Microsoft Windows NT Cluster Server: Programming and Administration* (<https://books.google.com/books?id=kUJnHJJlnpUC&pg=PA4>). CRC Press. p. 4. ISBN 978-1-4200-7548-9.
2. Mike Ebbers; John Kettner; Wayne O'Brien; Bill Ogden (2012). *Introduction to the New Mainframe: z/OS Basics* (<https://books.google.com/books?id=c-a1AgAAQBAJ&pg=PA96>). IBM. p. 96. ISBN 978-0-7384-3534-3.
3. Chip multiprocessing (<http://www.computerworld.com/hardwaretopics/hardware/desktops/story/0,10801,54343,00.html>)
4. <http://www.yourdictionary.com/multiprocessor>
5. <http://www.thefreedictionary.com/multiprocessor>
6. Irv Englander (2009). *The architecture of Computer Hardware and Systems Software. An Information Technology Approach*. (<http://www.wiley.com/college/englander>) (4th ed.). Wiley. p. 265.

7. Deborah Morley; Charles Parker (13 February 2012). *Understanding Computers: Today and Tomorrow, Comprehensive* (<https://books.google.com/books?id=-2Ewg8QX8U4C&pg=PA183>). Cengage Learning. p. 183. ISBN 1-133-19024-3.
8. Shibu K. V. *Introduction to Embedded Systems* (<https://books.google.com/books?id=8hfn4gwR90MC&pg=PA402>). Tata McGraw-Hill Education. p. 402. ISBN 978-0-07-014589-4.
9. Ashok Arora (2006). *Foundations of Computer Science* (<https://books.google.com/books?id=CrcoszZBMowC&pg=PA149>). Laxmi Publications. p. 149. ISBN 978-81-7008-971-1.
10. Ran Giladi (2008). *Network Processors: Architecture, Programming, and Implementation* (https://books.google.com/books?id=_7aH_4axpwAC&pg=PA293). Morgan Kaufmann. p. 293. ISBN 978-0-08-091959-1.
11. Sajjan G. Shiva (20 September 2005). *Advanced Computer Architectures* (<https://books.google.com/books?id=DhdCwk5AhbEC&pg=PA221>). CRC Press. p. 221. ISBN 978-0-8493-3758-1.
12. Multiprocessing (<https://archive.org/details/byte-magazine-1985-05>), BYTE magazine Volume 10, Number 05 (May 1985), p. 169.

Retrieved from "<https://en.wikipedia.org/w/index.php?title=Multiprocessing&oldid=779621528>"

Categories: Parallel computing | Classes of computers | Computing terminology

- This page was last edited on 10 May 2017, at 00:09.
- Text is available under the Creative Commons Attribution-ShareAlike License; additional terms may apply. By using this site, you agree to the Terms of Use and Privacy Policy. Wikipedia® is a registered trademark of the Wikimedia Foundation, Inc., a non-profit organization.