The InterNIC managed registrations of second-level domain names under the following top-level domains: edu, gov, com, net, and org. It also assigned blocks of Internet Protocol (IP) addresses to various entities, for example, to **Internet service providers**, who then distributed these IP addresses to their customers. Regional network information centers such as APNIC (Asia-Pacific Network Information Center) and RIPE (Regional Internet Registry for Europe) also received blocks of IP addresses from the InterNIC for distribution in their regions.

In October 1998, the U.S. Department of Commerce (DoC) and NSI amended their cooperative agreement, under which NSI had been the sole registrar and registry administrator for the .com, .net, and .org top-level domains. In November 1998, the DoC identified the **Internet Corporation for Assigned Names and Numbers** (ICANN) as the entity to oversee the transition to competition under the Shared Registration System (SRS). On 28 September 1999, the DoC, NSI, and ICANN reached an agreement that, in effect, meant the end of InterNIC as an organization, the liberalization of the DNS registration, and the recognition of ICANN as the nonprofit, private-sector corporation that would coordinate a select set of the Internet's technical management functions that were originally performed by the U.S. government or its contractors and volunteers.

FURTHER READING

Gates, Dominick. "Culture Clash in the Root Zone: Should Jon Postel Be the Master of Your Domain?" *PreText Magazine*, Mar. 1998. *http://www.pretext.com/mar98/features/story4.htm*

Hafner, Katie, and Matthew Lyon. *Where Wizards Stay Up Late: The Origins of the Internet.* New York: Touchstone Books, 1998.

Kujubu, Laura. "InterNIC Under Fire." *InfoWorld*, 5 Jan. 1998.

———. "NSI Protests Criticism of InterNIC System." *InfoWorld*, 12 Jan. 1998.

Postel, Jonathan. "Proposal for the Internet Corporation for Assigned Names and Numbers (ICANN)." National Telecommunications and Information Administration, 2 Oct. 1998. *http://www.ntia.doc.gov/ntiahome/domainname/proposals/icann/icann.html*

—*Manuel Sanromà*

Interpreter

See Compilers and Interpreters.

Interrupt

An interrupt is a disruption in the normal course of execution of a program. The computer stops the running program, gives control to an interrupt-handling routine, and returns to the program, after the interrupt has been serviced.

For example, a key pressed at the keyboard generates an interrupt to the processor, the processor stops the program being executed, goes to the keyboard-handling routine, stores the code of the character typed, and returns to the original program. An internal timer can also interrupt the processor periodically and the processor can keep track of the date and time. To the user, it is as if several processes were running in parallel—the keyboard routine, the timer, and the program—although only one process is running at any given time.

There are both hardware and software interrupts. The former are generated by any peripheral waiting to be attended. In the early days of computing, when input was needed from the computer's memory, the keyboard, or any other input/output device, the processor would stay in a loop asking the device if it was ready to send or receive data. This is called *polling*, but it was very inefficient. In modern machines, interrupts are sent through the bus, or main communication pathway.

Hardware interrupts can also be caused by exceptions—that is, when the execution of an instruction leads to an undefined operation. If, for example, the processor tries to read from a nonexistent address, an interrupt will be generated. The same happens if the processor tries to divide zero by zero. In some cases, the processor can decide whether or not it wants to service the interrupt. If the interrupt can be ignored (i.e., "masked") it is a *maskable interrupt.* If the interrupt cannot be ignored (a power-down interrupt, for example) it is a *nonmaskable interrupt* (NMI). The bit or bits that are set in order to ignore or accept the interrupt are called the *mask*.

Software interrupts are produced by special instructions that abandon the program being executed and jump to user-specified routines. This is done in the **per-**

sonal computer (PC), for example, through the INT instruction, which provides a method to call system routines. The system routines are burned in the BIOS ROM chip of the system and handle low-level operations such as accesses to the disk, to the internal timer, and others. Sometimes, deterministic interrupts are called *traps*. It is said of the instructions that invoke some service that they trap to the operating system.

Interrupts can be prioritized. If certain interrupts are more important than others (e.g., because a peripheral can wait), they must be serviced first. If an interrupt is being serviced, a higher-priority interrupt can stop the interrupt handler for that interrupt, be serviced, and when done, return to the first interrupt handler. Therefore, interrupts are handled in the sequence of their respective priorities.

The latency of an interrupt is the time it takes for the interrupt to be serviced. This can be a very important factor in **real-time systems**, that is, when a time bound has to be imposed on the service time. Some operating systems, notably **Unix**, do not provide an upper bound in the latency of an interrupt and cannot be used in mission-critical applications. Real time variations of Unix have been developed exclusively for this purpose.

Hardware vendors who build expansion cards for IBM-compatible PCs have to interface the cards to the rest of the machine. This is done through the expansion bus. The original IBM PC provided 16 Interrupt Request (IRQ) lines to signal the processor whenever a card wanted attention. The processor knows when an interrupt arrives and also knows which peripheral is requesting it. It can then decide if it should service it immediately or if it can wait. Expansion cards should therefore use different IRQs. Collisions of boards, preset to the same IRQ, were very common in the early PCs. However, the PCI bus, standard now in many PCs, allows IRQ lines to be shared and solves the collision problem. Also, the plug and play (PnP) system introduced by Intel and Microsoft is a way of configuring expansion cards so that they will not conflict with each other.

In general, interrupts are a way of organizing a computer system so that it can commit its processing resources to events that occur asynchronously, that is, nondeterministically.

FURTHER READING

Brown, Ralf, and Jim Kyle. *PC Interrupts: A Programmer's Reference to BIOS, DOS, and Third-Party Calls.* Reading, Mass.: Addison-Wesley, 1994.

Tanenbaum, Andrew S. *Structured Computer Organization.* Upper Saddle River, N.J.: Prentice Hall, 1990.

—*Raúl Rojas*

Intranets and Extranets

Intranets are organizational information systems based around many of the techniques and applications used for the **Internet**. They are not part of the Internet, as they are protected by security systems. Intranets make use of **interfaces** and data communication techniques originally developed for the Internet and the **World Wide Web** and are often a key aspect of both information management and knowledge management. From the late 1990s, many organizations with intranets also began developing **extranets**, which open up some parts of the intranet to people outside the organization. Extranets are now a central feature of **electronic commerce**. As organizations implement intranets and extranets, they often made dramatic changes to their organizational structures.

Intranets are such a recent innovation that they have received little treatment from historians of computing and information technology (IT). However, it seems likely that the concept of intranet was first coined by Steven Telleen (1947–), of Amdahl Corporation, in January 1995. He developed a methodology for implementing a new information framework for organizations, which he called the *IntraNet methodology*. At this time many organizations had experienced a number of major IT-related organizational transformations, with significant changes in the functional and operational units. There had also been significant downsizing, with the loss of many middle-management positions, and many employees were experiencing fatigue from the pace of IT-related organizational change. In this climate it was proving extremely difficult to implement further restructuring of IT systems and architectures. The IntraNet methodology, and the subsequent development of intranets, explicitly addressed many of these organizational issues.