ing under J. R. Womersley (who headed a new Mathematical Division) and serving briefly as assistant to **Alan Turing** (1912–54). The NPL was a government laboratory largely devoted to metrology; however, in the post-1945 years it was also seen as a center for research in computing. An early stored-program computer, the Pilot ACE, was completed there by 1950. In the early 1950s, Davies used the Pilot ACE to simulate road traffic and worked on the ACE computer design.

Following an NPL symposium on the mechanization of thought processes in November 1958, Davies' attention turned to the machine translation of languages, especially Russian–English, which was of Cold War interest. Through contact with Harvard University, a magnetic tape of a Russian–English dictionary was obtained (although inevitably in the incompatible **UNIVAC** rather than Pilot ACE format) and the first acceptable machine translations achieved in 1963. Davies was a technical manager for the U.K. government's Advanced Computer Techniques Project (ACTP) and served briefly at the Ministry of Technology in 1965.

In 1966, Davies became superintendent of the Autonomics Division (renamed the Division of Computer Science in 1968). He was then in a position to oversee a project based on one of his own key ideas. When communicating over a network, there are two options: Either, as is the case in a telephone conversation, the entire length of the connections can be kept open and dedicated to the duration of the communication (known as *circuit switching*), or the message can be passed from node to node. If a long message is divided up into smaller parts, which are then fed from node to node according to gaps in the traffic, this second method of communicating is known as packet switching. An early version of packet switching was used in telegraphy and had been proposed (but not implemented) for computer data networks by **Paul Baran** (1926–) of the RAND Corporation in 1961.

Davies independently proposed packet switching in 1965, and after a presentation by NPL staff at an Association for Computing Machinery meeting in Gatlinburg, Tennessee in 1967, it was adopted by **Lawrence G. Roberts** (1938–) for ARPANET. Davies envisaged a two-level data network: one at the national level connected to other networks at the second level of local organizations or buildings. A local area network around the NPL was working by 1970, using a **Honeywell** DDP-516 as a message-switching computer, with software written by Peter Wilkinson. Davies continued to direct the Computer Science Division until 1978, after which he worked in a number of areas, notably data security.

BIOGRAPHY

Donald Watts Davies. Born 7 June 1924 in Treorchy, Wales. B.S. (with honors) in physics from Imperial College, London University, 1943; B.S. (with honors) in mathematics from Imperial College, London University, 1947. Worked at the National Physical Laboratory, 1947–84, except when serving briefly as Ministry of Technology, 1965. Headed the Division of Computer Science, 1966–78; appointed Deputy Chief Scientific Officer, 1978. Private consultant, 1984–present. Recipient of numerous honors and awards, including the John Player Award, British Computer Society, 1974; distinguished fellow, British Computer Society, 1975; and John von Neumann Award, 1983; appointed as Commander of the British Empire, 1983, and as a fellow of the Royal Society, 1987.

SELECTED WRITINGS

Davies, D. W. "Communication Networks to Serve Rapid-Response Computers." *Proceedings of IFIP Congress 68*, Aug. 1968, pp. 650–656.

———. "The Principles of a Data Communication Network for Computers and Remote Peripherals." *Proceedings of IFIP Congress 68*, Aug. 1968, pp. 709–714.

Davies D. W., and W. L. Price. *Security for Computer Networks.* Chichester, West Sussex, England and New York: Wiley, 1984; 2nd ed.,1989.

FURTHER READING

Brandel, Mary. "The Promise of Packet Switching." *Computerworld*, Vol. 33, No. 18, Apr. 26, 1999.

Campbell-Kelly, Martin. "Data Communications at the National Physical Laboratory (1965-1975)." *IEEE Annals of the History of Computing*, Vol. 9, 1988, pp. 221–247.

Yates, David M. *Turing's Legacy: A History of Computing at the National Physical Laboratory, 1945–1995.* London: National Museum of Science and Industry, 1997.

—*Jon Agar*

dBase

dBase (versions II, III, and IV) was the most popular database system for personal computers before **Microsoft** and **Oracle** started dominating this mar-

ket. It was sold by Ashton-Tate, a company acquired by Borland in 1991.

dBase started around 1979 as the pet computer project of Wayne Ratliff (1946–), a programmer at the Jet Propulsion Laboratory in Pasadena, California. Ratliff wrote a program for his own personal computer using the relational approach, in which information is stored in tables. His first implementation was a port of a public domain system, called JPLDIS, from a **mainframe** to a **microcomputer**. He then started selling his database software, called Vulcan, by mail order. An entrepreneur, George Tate, convinced Ratliff to sell the product exclusively through the company he had just started, Ashton-Tate (the name *Ashton* was picked to make the name of the company sound serious). Since another company had a claim to the Vulcan name, the database was renamed *dBase II*. It was sold for several hundred dollars. The **CP/M** version of dBase II was released in 1981 and became an immediate success.

When the **IBM** personal computer (PC) was introduced, a new version of dBase was ported to this machine. *dBase III* was a more powerful product, capable of handling large data files. It was still very easy to program, and third-party vendors introduced **compilers**. dBase III was originally an **interpreter**: The system had to be started using typed commands, the user database was opened, and a user program could then be run. The program source code was always visible. However, developers wanting to preserve the confidentiality of their programs could compile them, transforming the source into an executable file. The most used dBase compiler was Clipper, introduced by Nantucket in 1985.

Being the first database product for microcomputers, dBase set a standard against which other products were measured. The heyday of dBase was the "before **Windows**" era. The user communicated with dBase using a command line in which such keywords as SELECT and PRINT could be typed. Since the data were stored in tables, the user would select records meeting some criteria and would then print those fields in which he or she was interested. dBase was an interactive product, but the user would learn to program simply by watching the result of his queries. A series of queries, stored in an ASCII file, was already a dBase program. dBase made every user a programmer.

dBase IV, introduced in 1988, made programming even easier. Most of the commands were started by clicking on buttons and query forms from the *Control Center*. The sequence of commands selected could then be stored in a program to be used repeatedly. Also, it was now possible to start SQL queries within dBase programs, expanding the universe of users who used the database. Input screens could be "painted" and did not have to be designed painstakingly using elementary commands. However, dBase IV never became as popular as dBase III, in part due to the many bugs in the first release. Ashton-Tate started losing ground in the database arena at about this time.

In 1990, while dBase users were still waiting for version IV 1.1., a federal judge ruled against Asthon-Tate in a lawsuit that the company was running against Fox Software and Santa Cruz Operations claiming copyright infringements of the dBase language. At issue was the "look and feel" of the dBase product, which Ashton-Tate claimed could not be copied. But the judge ruled that Ashton- Tate had no rights over the query language, since dBase had originated from a public domain system (JPLDIS) and this has not been made clear in the company's patent filings. Although the ruling was later reversed by the same judge, Ashton-Tate continued its downfall and the company was bought by Borland, which in July 1991 was already working on a database for Windows for U.S.$440 million.

In 1994, Borland released *dBase 5.0* for Windows, the first dBase version with a GUI. But now there were more competitors, such as Microsoft's Access, and the new dBase could not recover its user base. Borland, which in 1998 changed its name to Inprise Corporation is the owner of the dBase brand. The latest version of the software is called *Visual dBase*.

The importance of dBase as a product relies on having brought database handling capabilities to microcomputers. In 1987, the fourth-most-sold program for the IBM PC was dBase III and in 1995, *Byte* magazine selected dBase as the fourth most important software product in history. Eventually, Ashton-Tate failed because it could not adjust in time to the new operating systems brought out by Microsoft and Apple.

FURTHER READING

Barnes, Lan. *Introducing dBASE II.* New York: McGraw-Hill, 1985.

———. *Introducing dBASE III.* New York: McGraw-Hill, 1985.

Masterson, Michael P., ed. *dBASE Language Reference with Annotations.* New York: Random House, 1993.

—Raúl Rojas

DBMS See Database Management Systems.

DB2

DB2 is a **database management system** (DBMS) sold by **IBM**, whose market share in this segment of the software industry was about 30 percent in 2000. DB2 runs on a variety of **operating systems**, including OS/2, **Windows** and **Windows NT**, variations of **Unix**, and IBM's OS/390 and OS/400. There are currently more than 3 million DB2 installations.

DB2 is a relational database—that is, data are stored in tables (*relations*) that can be queried using a special language such as **SQL**. This language was developed by IBM and in the late 1970s was put into the public domain. The first relational database sold by IBM was SQL/DS, released in 1981, which was replaced two years later by DB2, a database for IBM mainframes. DB2 has continued to evolve since its introduction, adding new features every year and also new variations of SQL. Being an SQL-based language, DB2 provides data definition, data manipulation, and a data control language.

DB2 is one of the few examples of software that moved successfully from the **mainframe** to smaller computers. IBM added the suffix UDB to DB2 to indicate that it is now a universal database. The key objective of IBM has been to position DB2 at the center of "complete solutions," by offering many other software products that access the DB2 server software. To implement this strategy, all DB2 groups within IBM were fused into a single group that wrote unified code for all platforms.

To couple the database with applications, DB2 provides a functionality that makes programming with this system easier. An interface exists for most common programming languages, so that DB2 queries can be started from within a program. Also, it is now possible to access DB2 databases through the **World Wide Web**.

The latest release of DB2 added support for data warehouses and online analytical processing (OLAP). IBM's intention is to provide *scalability*—that is, the capability of greatly increasing the size of a database—and *extensibility*, which refers to interfacing future applications to DB2. This is the case of electronic commerce on the Web, for which no definite standards exist.

The largest producers of database systems are IBM, **Oracle**, **Microsoft**, Informix, and Sybase, in that order. Although IBM had lost market share in the 1990s, it became the leading database vendor after switching to its "universal" strategy. The database market is expected to grow to U.S$10 billion by 2003, and DB2 is expected to continue to be one of the principal DBMS products.

FURTHER READING

Lawson, Susan, and Richard Yevich. *DB2 High Performance Design and Tuning.* Upper Saddle River, N.J.: Prentice Hall, 2000.

Mullins, Craig. *DB2 Developer's Guide.* Carmel, Ind.: Howard W. Sams, 1992; 4th ed., Indianapolis, Ind.: Macmillan Computer Publishing, 2000.

—Raúl Rojas

Debugging

The word **bug** as a synonym for a computer programming error has been in use for more than 60 years. Debugging refers to the process of finding and removing errors in programs.

Usually, the programmer examines a program's source code, trying to identify the offending portion. In more complex situations, a debugging tool allows the user to go step-by-step through a program, setting breakpoints at specified statements and examining the values of variables. The first debuggers worked only with **assembler** language, but there are modern debugging tools for high-level programming languages which interpret a program line by line. The debugger for the language **Prolog**, for example,