Introduction

Databases and systems for their management are today more and more important for individual and corporate applications. Furthermore, the spreading of the Internet has made possible the achievement of centralized information systems, accessible from everywhere on the Net, both for querying data and for managing them.

A special role is played in this process by open source software and especially by some packages which can be linked all together to build online information systems.

It is well known that open source tools are today widely used by developers and programmers and that they are a valid and reliable alternative to proprietary software, but often a mortal terror prevents common people from using them.

In what follows, after a short survey of historical events leading to the creation of the most famous open source packages and the Open Software Foundation, a description of the author’s experiences is proposed, so showing how, in some special cases, an effort in learning new topics and developing adequate skills can produce relevant effects in solving very different problems and in researching.

Background

In the late ’60s the spreading of computing systems led many scientists to hypothesize that the target of shared computing and the consequent access to computer resources by everyone could be hit with the use of great centralized systems, equipped with multi-program, timesharing and multiple-access operating systems. The MULTICS (Multiplexed Information and Computer Service) case is perhaps the best example in this respect; it was a project carried out by MIT, Bell Labs, and General Electric, which meant creating a huge machine providing computing power to everyone (Corbato, Saltzer, & Clingen, 1972). Notwithstanding the MULTICS project was abandoned, it introduced many seminal ideas in computing literature, and some computer scientists at Bell Labs worked on it and made a one-user version of the system for a PDP-7 (DEC minicomputer); they called this new system UNICS (Uniplexed Information and Computing Service) but suddenly renamed it UNIX (Bach, 1986). It is well known that Ritchie’s development of the C language, the rewriting of the whole UNIX system in this new language, and its distribution for free to universities and research centers made the fortune of this operating system, which was implemented and installed on a great variety of computing systems in a few years.

The development of LSI (large-scale integration) circuits in late ’70s led to personal computing, i.e., to computers not very different for their architecture from minicomputers but more and more cheaper. Everyone could now have a computer for his/her own use and, as history demonstrates, the dream of great centralized systems for shared computing was abandoned.

The case of operating systems, in the author’s opinion, is emblematic for the impulse that personal computing gave to proprietary software. It is well known, in fact, that two main operating systems became popular on PCs. The first one, MS-DOS by Microsoft Inc., for the IBM PC and all machines equipped with the 8088 CPU, and the second one, UNIX by various distributors, for high-level personal computers equipped with the Motorola 68000 CPU family (Tanenbaum, 1987); none of the above systems was for free or was freely available. In other words, the introduction of PCs didn’t help computer science knowledge, until then mostly shared among scholars and researchers, to exit from laboratories, and only the efforts of a few people led to the creation of operating systems freely available with their source code.

Once more, UNIX is the reference example. Its transformation into a commercial product with a license prohibiting the public study of its source code led many universities (adopting it for their operating systems courses when it was free) to abandon the system. Some scholars, on the contrary, decided to entirely rewrite the kernel of the system, letting it open, and one of the most famous examples in this regard was MINIX, developed by A. S. Tanenbaum (Tanenbaum, Van Staveren, Keizer, & Stevenson, 1983).

Autonomously from Tanenbaum, Richard Stallman and Linus Torvald developed prototypes of freely available operating systems and decided to join their efforts for creating GNU/Linux (today better known as Linux); it was
available on the Net in 1991 (Beck et al., 1996), but in a few years it evolved (and is still evolving) and became so steady and reliable as to be a serious and valid alternative to Microsoft Network Server’s software.

Operating systems were not the only software freely available or developed to be freely accessible with their source code, but the success of Linux (as an operating system) is very important for the spreading of many other initiatives (which used that OS for their implementation).

Faster and more efficient ways for accessing software became available with the Internet, and the use of open and/or free software (under special licenses like BSD, GPL, etc.) was made easier. New individuals and communities of developers worked on other software projects and adopted the same strategy of making freely available their source code; consortia like FSF (Free Software Foundation) and OSF (Open Software Foundation) were then created for helping people in defining standards, protecting their rights, and continuing the hard work of developers.

Among the various projects one can find on the Net, the following ones, having a relevant part in what follows, will be analyzed in greater detail: the Apache Web server, the PHP scripting language, and the PostgreSQL RDBMS.

The Apache Web server was developed by a group of scientists who left the NCSA System Development Group and was made freely available on the Net (the Apache Web site http://www.apache.org/ is a good starting point for the downloading of the server software). Once compiled and the program started, the HTTP daemon looks for requests coming from the Net and creates system processes to answer to them. One of the main features of this Web server is its modularity and the chance for a Web server administrator to integrate special modules, enhancing the server functionalities within it.

PHP is a scripting language (the reference Web site is http://www.php.net/), making easy for Webmasters the creation of Web interactive pages (i.e., FORMS letting data go back from client to Web server). Very valued features of this software are its modularity, its embedding features, and the interaction it guarantees with most widely used RDBMSs.

The PostgreSQL RDBMS is a software tool for the management of tables and queries in a relational database (it is available from the Web site http://www.postgresql.org/) by means of the well-known SQL (Structured Query Language), granting special users an easy access to data. The project for this software was firstly developed at Berkeley when the first examples of DBMSs (database management systems) were analyzed and discussed and the relational model was compared with the hierarchical and network models and adopted for it (Bracchi, Martella, & Pelagatti, 1987).

From the remarks reported until now, it can be easily deduced that the Internet and the above tools made easy and cheap the creation of information systems accessible by general and special users for querying and managing the databases hosted on Web servers (Cartelli, 2004a). In other words a Web server on the Internet (now mostly a PC) with all the above tools installed and running and the right Web pages for the storing/retrieving functions accessing a database becomes very similar to a mainframe with its virtual terminal services.

OPEN SOURCE SOFTWARE AND EDUCATION: TWO CASES

Many sites on the Net today allow access to open source software, and some among them host whole projects, already developed or still evolving, with their communities of developers. Users, scholars, and programmers can easily download from these sites the software they need and simply use it or can take part in their developmental projects. An example of the above sites, SourceForge (http://www.sourceforge.net/) is reported here.

Nevertheless one can be induced to create new applications if the already existing ones seem not adequate for special situations or particular problems.

The experiences reported below are good examples, in the author’s opinion, of the need for planning and creating special information systems by means of the open source software: The first one concerns the instruments to be used for paleographic research and teaching, and the second one concerns the carrying out of a special e-learning platform. Both of them are based on the use of the Linux operating system, the Web server Apache, the PHP language, and the PostgreSQL RDBMS.

Open Source, DBMSs, and the Community of Paleographers

The two information systems described here are the result of the author’s cooperation with M. Palma, a professor of Latin paleography at the University of Cassino (Italy). The first system is devoted to the management of the data concerning women who wrote manuscripts in the Middle Ages (women copyists); the second one manages the bibliography of the manuscripts written in Beneventan, i.e., an ancient medieval script used in South Italy.

The main aim of the dynamic Web site (http://edu.let.unicas.it/womediev/) named Women and Written Culture in the Middle Ages (Cartelli, Miglio, & Palma, 2001) was to systematize the data emerging from the research on women copyists while leading to an instru-