Open Source Software

As the internet bubble approached bursting point, voices arose against manufacturers who entrenched their software products within monopolistic bastions; in particular the Microsoft (quasi) monopoly became the target of verbal attack and lawsuits. At the turn of the millennium Microsoft controlled over 80% of the market in operating systems and 90% of the business-applications market. Nevertheless, the company denied anti-competitiveness, arguing that the software market had benefited greatly from its efforts leading to standardisation and relatively low-cost software for the millions.

Escaping Monopoly

Monopolisation is inherent to the capitalist system; todayâ€[™]s many takeovers, whether mutually agreed or hostile, are witness to the eagerness of companies to eliminate competition. Indeed, so keen are they in this regard that government institutions are needed to inhibit monopolisation, a task they carry out with oscillating success. Resistance may also emanate from the user, particularly when it dawns on him that there is only one game in town and the manufacturer has taken him hostage. In the wake of the stunning tumble in the share values of dot.com companies, the general public revealed a broad interest in open-source software. Press coverage of the phenomenon was dominated by bewilderment at the emergence of such a thing as â€[°] freely distributedâ€[™] software, and software engineer Linus Torvalds, originator of Linux, became an icon. This operating system, around since 1991, was an unparalleled success thanks to its openness. Users, themselves developers, fix bugs and adapt the system to new innovations. Improvements are made public and rapidly assimilated into the next official release of Linux Kernel. Non-technical users benefit by participating in the many internet discussion forums where answers may be got to questions on use and possible malfunctioning.

Share and Share Alike

Most commercial software is made available in executable code consisting of ones and zeroes, the only language computers understand. By contrast, open-source programs always comprise the source code from which the executable code is compiled. Written in C++, Java, and the like, the code is accessible and comprehensible to any programmer. Although the press around 2000 reported on open-source software as if it was something astoundingly new, free exchange of source codes is as old as the place of computers as commodities in universities and research institutes. Indeed, from the very outset it was the policy of both academic institutions and commercial research centres that programmers unrestrictedly distribute and share the products of their intellectual efforts. Just as scientists publish their research results, software was considered a research product free for everybody, and by giving it away the scientific programming community hoped that others would use and improve on it. These researchers even considered the free distribution of software a prerequisite for cumulative furthering. Treating software as a tradable asset was far from anyone's mind and the rapid sharing of technologies led to huge progress, the sweet fruits of which we are all enjoying today.

Meaning of Freedom

And now, after 25 years of proprietary software and in a world in which the paradigm of capitalism has been hoisted to the zenith, giving away software for free is again increasing in popularity. Why should somebody give freely away that which he has laboured long and hard over †Freeâ€[™] is here not synonymous with †gratisâ€[™] or †without obligationâ€[™]. Open-source software is distributed within the framework of conventional copyright law, in which ownership is asserted and exercised. Use requires a licence providing a number of permits and constraints. The licence may come gratis or be paid for by fee; †freeâ€[™] here means freedom of use, not price. Why give software away For the individual developer there is the excitement of mind-share: working with people around the globe who appreciate his or her software. Throwing software into the open-source ring also brings with it the possibility of fame, and maybe even fortune: a well-paid job in Silicon Valley. There is also the ideological motive conveyed by the free-software movement active since 1984 and driven by Richard Stallman, instigator of the GNU project. In proclaiming the essential contribution made to the development of computer science by freely available code, one is also recognising the need for measures to prevent pirates from making a profitable business from other peopleâ€[™]s efforts. The answer to this threat was the GNU General Public License (GPL), which allows anyone at will to copy and distribute software licensed under the GPL, provided that they do not restrain others from doing the same, either by charging or restricting through further licensing. The GPL also requires works derived from work licensed under the GPL too to be licensed under it.

Winners and Losers

Open-source software enables small, innovative enterprises to enter established markets by introducing types of service the core of which no longer consist of the software itself but of a bundle of services packaged such that the final result interests the consumer. And part of that bundle may be in-house developed software, thrown into the open-source ring so that others may not only use but also improve on it. A main incentive for placing stuff in the ring is the conviction that open-source software has greater integrity and security than closed, proprietary products because many more eyes are scrutinising the code. Large corporations have much to lose from the current shift in interest from proprietary to open-source software.