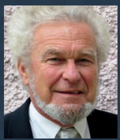
50 & 25 YEARS AGO



EDITOR **ERICH NEUHOLD**
University of Vienna
erich.neuhold@univie.ac.at

JANUARY 1969

In this issue, two time-sharing systems of that era are discussed. They are mostly interesting for explaining the technology that existed at that time and so we can see the tremendous progress we have made since then.

Time Sharing and its Applications (p. 3) "Because the time-sharing computer is capable—unlike the standard computer—of working numerous problems at the same time, each user seems to have the computer exclusively to himself. With this direct access to the computer, each user is able to carry on a 'running dialogue' with the computer. He can alter, add to, or even enter a new program instantly, because he is dealing directly with the computer. Com-Share Southern (CSS) offers a sophisticated combination of hardware and software to time-sharing customers. This, coupled with an extensive research and development program and the ability to provide local dialing access to a computer nationwide provides a user with a uniquely capable system. This article describes time sharing capability as offered by Com-Share Southern." (p. 4) "Each user, when 'online' with the computer, is allotted up to 16 K of memory for the execution of his program. Memory is automatically apportioned by the system monitor on an as-needed basis in units of 2 K. The monitor is the computer's control system. It determines in which sequence programs will be executed, how much memory is given to each user, and does all of the governing and bookkeeping for the system. … The most complicated question is that of 'how long should a user execute' once he is activated. A user who is editing his program would want a quick response from the terminal (short time slice) but a user who is executing a program would want a long time slice for execution. CSS's system tunes itself to a user's demand and gives each user the time slice (quantum) determined by his demands on the system." (p. 6) "Data and program security are of primary importance on a computer system which serves multiple users. CSS guarantees confidential and safe storage of user information in a number of ways … account numbers and passwords … control characters … answer back drums." *[Editor's note: The system works with a 64-K character core and two 2-Mill character fast drums! However, it allows multiple users shared access to a computing resource, eliminating the sequential processing usually available at that time.]*

A Practical Time Shared Computer System (p. 9) "With the growth of computers in speed, power and price, 'time-sharing' has received increasing acceptance as a means of making the power of a large and expensive computer available online, to individual users, efficiently and economically. … The development project for the Hewlett-Packard Model 2000 A Time-Shared BASIC System was initiated with the objective of developing a system to fulfill economically the majority of a user's needs, while leaving the more complex and expensive functions to be satisfied by larger systems. … The time-sharing system is built around the HP Model 21168 Computer with a 16-bit word length (plus parity) and 16,384 words of magnetic core memory. For bulk high speed memory, the system uses a magnetic disk memory with 348,160 words of storage. Disc storage can be expanded by adding up to three more disk-units providing 1.25 million words of program storage." *[Editor's note: The system actually represents, following the tradition of Hewlett Packard at that time, a midrange system with a price tag of approximately US$140,000.]*

Prison Rehabilitation Through Education (p. 15) "How can we, as members of the computer 'fraternity,' assist some of these men and women and make them useful members of our society? First, we must understand the cause of recidivism—the reason that parolees return to their old ways and hence back to prison. As a parolee so aptly put it—serving time is a brain-numbing treadmill. … Walpole is the maximum security prison in Massachusetts and better than 90% of the population are high school drop-outs. … We told them that we would start off by teaching them the Fundamentals of EDP, but that this was background material only—it would not help them get a job on the outside. However they must successfully pass this course before going on to programming. We also told them they must be willing to teach other inmates, and that the success of the program was strictly

Digital Object Identifier 10.1109/MC.2018.2890189
Date of publication: 11 March 2019

 0018-9162/19©2019IEEE

up to them." (p. 16) "After correcting their scores, we found that 20% of these men scored as high as the average college graduate. Since the original group, we have tested about 70 more men and found that these averages have pretty well held up." (p. 17) "We asked our operating system expert to present a series of lectures on operating systems, their use, how they work, how all software ties in to them. These lectures took 10 sessions. We noticed an interesting phenomenon during this time. The men would not ask questions during class, and we found out later that the reason for this was to get as much information as possible. Then they would spend the next 2 or 3 days talking over what each one had heard. Thus, no one man heard or understood all the instructor had said, but by pooling their knowledge they covered all the information. … Up to this point in time, we have trained, including both our classes and those taught by the inmates, approximately 10% of the population at Walpole. By training inmate teachers, we know this will continue to be a major area of education and rehabilitation in this prison."

Trends in Aerospace Digital Computer Design (p. 18) "This article summarizes some of the recent trends in aerospace computer design that can be inferred from the changing characteristics of computers developed during the period 1962–1968. In an earlier paper (IEEE Computer Group News. January 1968), characteristics of 40 computers developed during 1962–1967 were described. … Characteristics of the sixty computers used as a basis for this paper are given in Table 1, where the computers are listed according to their date of introduction." *[Editor's note: The listing is extensive and provides many of the properties of the individual systems described. However, unfortunately, the article does not discuss what essential properties distinguish those computers from the computers utilized elsewhere, be it research or industry.]*

JANUARY 1994

www.computer.org/csdl/mags/co/1994/01/index.html

Industrial Computing: A Grand Challenge (p. 12) "Computing is playing an increasingly important role in industry. Example applications include high-speed transportation systems, advanced manufacturing plants, air-traffic control systems, and high-speed mass-transit systems. Power generation and distribution, advanced medical equipment, and defense systems are other examples." (p. 13) "Industrial computing system requirements for correctness, performance, and human-computer interfaces are similar to those for business and scientific computing systems. In addition, industrial computing systems are often distributed in nature and have distinctively stringent timing, reliability, and safety requirements imposed by the application environment. … The creation of a renewable industrial computing architecture and transitioning it into practice is a Grand Challenge for computing in the nineties. This special issue of *Computer* is a first step in responding to this challenge, addressing a ubiquitous aspect of industrial computing: meeting real time and fault-tolerant requirements."

TTP—A Protocol for Fault-Tolerant Real-Time Systems (p. 14) "The Time-Triggered Protocol (TTP) we present here is an integrated communication protocol for time-triggered architectures. It provides the services required for the implementation of a fault-tolerant real time system: predictable message transmission, message acknowledgment in group communication, clock synchronization, membership, rapid mode changes, and redundancy management. It implements these services without extra messages and with only a small overhead in the message size." (p. 16) "We intended TTP for Class C safety-critical automotive applications: real time control activities requiring guaranteed timeliness and fault tolerance. Our objective was a protocol that integrated all services needed in such applications: —Message transport with predictable low latency—Fault tolerance—Temporary blackout handling—Clock synchronization—Membership service—Distributed redundancy management—Support for rapid mode change—Minimal overhead—Utmost flexibility without compromising predictability." (p. 17) "TTP is an integrated protocol that provides these services without the strict separation of concerns proposed in the layered Open Systems Interconnection model. The OSI model is excellent conceptually for reasoning about different design issues. However, timeliness is not one of its goals, and it is inadequate as an implementation model for time-critical protocols." (p. 22) "TTP also scales well to high transmission speeds for fiber-optic systems since it requires no bitwise arbitration. … We therefore estimate that the complexity of a TTP controller chip with the two redundant I/0 channels is less than 100,000 transistors, excluding the memory. Because the present level of VLSI integration is far beyond 1 million transistors, it seems technically possible to integrate a TTP controller into a single-chip microcomputer." *[Editor's note: The chip development following these ideas was actually very successful and is still present in the 1998 spinoff company TTTech of Austria.]*

Rate-Monotonic: Analysis for Real-Time Industrial Computing (p. 24) "Rate-monotonic analysis, a collection of quantitative methods, provides a basis for designing, understanding, and analyzing the timing behavior of real time industrial computing systems. … One guiding principle in real time system resource management is predictability, the ability to determine for a given set of tasks whether the system will be able to meet all of the timing requirements of those tasks. Predictability calls for the development of scheduling models and analytic techniques to determine whether or not a real time system can meet its timing requirements." (p. 26) "A comprehensive scheduling theory has been developed for fixed priority preemptive systems based on the rate-monotonic algorithm; the theory is called generalized rate-monotonic analysis." (p. 32) "There have been many extensions of the rate-monotonic-scheduling algorithm to address a multitude of other practical problems that arise with real time systems. … Our own future research will include integrating these techniques for timing analysis with dependability techniques to achieve safe operation, high availability, and online software/hardware upgrades."

The Timed-Token Protocol for Real-Time Communications (p. 35) "The timed-token protocol is a token-passing protocol in which each node receives a guaranteed share of the network bandwidth. Partly because of this property, the timed-token protocol has been incorporated into a number of net-work standards, including the IEEE 802.4 token bus, the Fiber Distributed Data Interface (FDDI), and the Survivable Adaptable Fiber-Optic Embedded Network (Safenet). Networks based on these standards are becoming increasingly popular in new generation real time systems." (p. 40) "Because the synchronous-bandwidth allocation method uses only information local to a node, we can create or modify synchronous-message streams locally without reinitializing the network. Furthermore, our parameter selection methods are compatible with current standards of the timed-token protocol such as IEEE 802.4 and FDDI."

Program Implementation Schemes for Hardware-Software Systems (p. 48) "For the most part, we can apply high-level synthesis techniques to synthesis of systems containing processors by treating the latter as a 'generalized resource.' However, the problem is more complex, since the software on the processor implements system functionality in an instruction-driven manner with a statically allocated memory space, whereas ASICs operate as data-driven reactive elements. Due to these differences in computational models and primitive operations in hardware and software, a new formulation of the problem of cosynthesis is needed." (p. 55) "Software component design for such systems poses interesting problems because serial program execution must interact with concurrent hardware operations. In our approach to system synthesis, we implement the software component as a set of program routines called threads. We preserve the concurrency inherent in the system model by interleaving the execution of threads. Further, dynamic scheduling of fixed latency threads avoids unnecessary serialization of operations in a graph model. However, even with the simplified target architecture, accurately characterizing and synthesizing the software component is challenging, and our work represents only a first step toward system software synthesis. We are now working to extend this model to include the effects of hierarchical memory schemes and the reduction of communication overheads by using channel-sharing and data-encoding schemes."

Asimov's Laws of Robotics: Implications for Information Technology—Part 2 (p. 57) "The zeroth law. After introducing the original three laws, Asimov detected, as early as 1950, a need to extend the first law, which protected individual humans, so that it would protect humanity as a whole. Thus, his calculating machines 'have the good of *humanity* at heart through the overwhelming force of the First Law of Robotics' (emphasis added). In 1985 he developed this idea further by postulating a 'zeroth' law that placed humanity's interests above those of any individual while retaining a high value on individual human life. —Law Zero: A robot may not injure humanity, or, through inaction, allow humanity to come to harm." (p. 62) "Modification of the laws, however, leads to additional considerations. Robots are increasingly required to deal with abstractions and philosophical issues. For example, the concept of humanity may be interpreted in different ways. It may refer to the set of individual human beings (a collective), or it may be a distinct concept (a generality, as in the notion of 'the State'). … In this era of powerful information technology, professional bodies of information technologists need to consider: identification of stakeholders and how they are affected; prior consultation with stakeholders; quality assurance standards for design, manufacture, use, and maintenance; liability for harm resulting from either malfunction or use in conformance with the designer's intentions; and complaint-handling and dispute-resolution procedures." *[Editor's note: The article, a continuation of the one in the December 1993 issue, further elaborates, based on Asimov's laws, the implications autonomous robots will have for humanity and our everyday life, discussions that are currently going on about autonomous cars, autonomous weapons, and many other autonomous things. Reading it may actually give us ideas about some of the issues currently discussed.]*

Open Channel: Submittal for a New Year's Resolution (p. 128) "We, the CS&E community, hereby resolve for 1994 to modernize our approach to technical meetings and to the peer review process for archival journals. The current system is faulty in emphasizing evaluation over constructive feedback and competition over advancement of knowledge. … Technical meetings. How pompous for a program committee to tell an author that a submitted paper is good but that it can't be accepted because it isn't in the top 10% of all submissions. A low acceptance rate doesn't mean high quality of accepted papers, partly because 10% of something that's 95% low quality is still at least 50% low quality—even with a perfect review process. … Archival journals. Journals that expect quality referee reports should pass every submitted paper through a swift preliminary review so that a paid editor can screen out clearly unacceptable papers on the basis of presentation quality and suggest improvements. Only after an author has written a readable paper should the valuable time of technical referees be requested. We also should resolve to use the Internet to streamline the review process with electronic interaction among authors and referees. Authors should be able to ask about and respond to referees' questions and should expect referees (still known only as A, B, and C if anonymity is required) to substantiate their objections in a timely and scholarly technical-exchange fashion." *[Editor's note: The article of 25 years ago clearly favors journal publications against conference proceedings by downgrading conferences to social meetings for exchanging research ideas even in unfinished form. Unfortunately, in my humble opinion, this thinking has been adopted by most of the research community and possibly also by most journal publishers. Even conferences seem to move to the journal first/conference second model, but, so far, I have not seen an increase in overall quality, either in conferences or in journals.]*