8

DANISH INTERLUDE 1984–87

Student democracy and teaching in Denmark – Danish industry uses Concurrent Pascal – Consulting for GN Corporation – Rocking the boat.

After fourteen years in America I suffered a first-class attack of homesickness for Denmark. I had written the book on operating systems that was my original reason for coming to the United States. And, through my work in operating systems and concurrent programming, I had fulfilled my dream of making fundamental contributions to a new field. As Americans say, "Been there—done that!" Now what?

At this point in my life, I longed to return to Denmark and continue the life I had left behind. If you think I was deceiving myself, why, you are absolutely right. But did I listen to reason? Nope, I just had to go back and discover for myself that my lost youth was indeed, well, lost. Milena, who had no wish to leave the United States, was surprisingly understanding. She told our children, Mette and Thomas: "Dad is unhappy. He has to get Denmark out of his system." Looking back, I find it unforgiveable that I turned their lives upside down, just as my son was close to graduating from high school, and my daughter was ready to start her university education.

In the fall of 1983, I applied for a new professorship in datalogy at the University of Copenhagen. Danish universities do not ask for confidential letters of recommendation from colleagues at other universities. Instead I was asked to submit copies of my best books and papers to an ad hoc committee consisting of professors Peter Naur and Peter Johansen from University of Copenhagen and Kees Koster from the Catholic University of Nijmegen, The Netherlands. After studying my work for four months, the committee submitted a six page summary of my career and scientific contributions recommending that I be appointed to the vacant professorship.

In an egalitarian society like Denmark, you were constantly aware, as Vartan Gregorian put it, "that people think they are equal, and that whether

Copyright © 2004 by Per Brinch Hansen.

or not that is accurate is irrelevant." All professors received the same salary independent of their achievements and the terms of their appointments were not negotiable.

Nevertheless, I had one concern before I was ready to accept a position in Denmark: At USC I had recently created a personal computer lab with 40 IBM PCs that enabled me to teach a course in which students wrote single-user operating systems in the programming language Edison. Would it be possible for me to establish a smaller PC lab at the University of Copenhagen?

Since I was used to negotiating directly with high-level administrators in the United States, I wrote a letter to Bertel Haarder, Danish minister of education, asking for his help in providing funds for a handful of IBM PCs. During a visit to Denmark, I also met him privately. Haarder, who after all was a politician, promised to find money for the PCs.

I soon learned that bypassing the normal channels of communication just isn't done at Danish universities. On April 27, 1984, the following item appeared in the newspaper Politiken (translated into English by me):

> *Gets job in spite of concerns.* This summer the Institute of Datalogy at University of Copenhagen (DIKU) will increase the number of professors from two to three. The favorite for the position is one of the leaders of the international world of computing, dr. techn. Per Brinch Hansen. He is currently professor of computer science at University of Southern California.
>
> The 45-year old Danish engineer is number one on the institute's confidential list of preferred applicants. . .However, it raised some concern at DIKU, when Per Brinch Hansen as a condition for his appointment practically demanded that the institute make six IBM computers of a specific type available. DIKU, which has one of them, now plans to acquire two more in the immediate future and three more later on.
>
> In 1978, the internationally known Danish computer scientist wrote—as the first in the world—a doctoral thesis about the special problems posed by computer programs that execute many tasks at the same time.
>
> Further down on the institute's list are people from DIKU's present staff.

Three months later, I received official notification that "We, Margrethe the Second, Queen of Denmark, by the grace of God, makes it known that We hereby, from August 1, 1984, appoint professor, doctor technices Per Brinch Hansen, who is a Danish citizen, as professor with permanent appointment.”

DIKU was a child of the student uprising in the late 1960s. For generations, every university department had been headed by a single professor who had the final say in all matters concerning curriculum, examinations, research, and appointments. Inspired by the student riots in the United States and France, Danish students and instructors staged demonstrations in the spring of 1968, shouting “Down with the tyranny of professors,” “Student participation NOW,” and “Research for the people.”

Everybody agreed that something had to be done. The politicians, who knew how to count votes, were well aware that the number of students far exceeded the number of professors. In May 1970, the Danish parliament passed a new Statute of Administration (“styrelsesloven”). From now on, every university department would be governed by a council (“institutråd”) with equal representation of teachers, students, and staff members.

In 1969, Peter Naur became the first Danish professor of computer science (“datalogy”). The following year, associate professor Edda Sveinsdottir became the first democratically elected head of DIKU. The new department couldn’t have made a better choice: Edda, who was friendly and helpful, attacked problems with boundless energy. Her innovative research in three-dimensional scanning of the brain (“computerized tomography”) and Peter Naur’s pioneering work in compiler development helped establish DIKU’s tradition as a center of applied computer science. In 1987, Edda became professor of datalogy at Roskilde University.

In my time, DIKU’s council consisted of sixty teachers, students, and staff members. Once a month, the council met for a couple of hours. Consequently, every teacher had only a couple of minutes to express personal opinions about departmental issues and had only one of sixty votes, independent of personal ability and achievements. Since teachers tended to vote as individuals, students often prevailed by voting unanimously.

When a group of people have the right to decide anything in democratic fashion, they naturally concentrate on problems they understand and ignore less familiar issues. At one departmental meeting, the discussion centered around problems with our printing office, the sun shades in the cafeteria, the reliability of the elevator, the use of office space, and the absence of teachers from these meetings “which they are obliged to attend.” Under the new system of governance, the focus of discussion was no longer innovative teaching and research, but job satisfaction.

My first impression was that a handful of teachers did interesting research, while the majority were unproductive. Looking through the Science Citation Index for the previous five years, I found that worldwide only one third of our faculty was cited in the works of other researchers.

As the Harvard dean, Henry Rosovsky (1990), has pointed out "Not everything is improved by making it more democratic:"

> The limiting case exists in some European universities where the practice of "parity" was born in the 1960s. Power over virtually all decisions came to be equally shared between students, faculty, and employees—and not infrequently the government. The educational results have been disastrous. Academic standards declined and a sense of mission was lost.

However, Edda Sveinsdottir felt that the endless discussions, that were necessary to resolve even minor issues, were worthwhile because they often led to decisions that everyone could live with. Personally, I found it pointless to listen to student representatives, who had never done research, but nevertheless believed that the problems of society could be solved by trying to control the unpredictable nature of research. If people had told Thomas Edison what to do, he might have invented a faster telegraph key instead of the electric light.

Needless to say, the reality did not always resemble the politically correct utopia. When I suggested to my colleagues that I would like to revise the operating system course completely, their immediate reaction was "Please don't! If we change even one course, the militant students will use that as an excuse to demand a revision of the entire curriculum." In the end, I simply had to accept that the Danes had developed an educational system that valued cooperation and peace of mind more than individual pursuit of excellence.

It was a joy to teach Danish undergraduates. It would, of course, be unfair to compare them to Caltech students. But, thanks to the excellent Danish high schools, they were, on the whole, better prepared than most American undergraduates at USC or Syracuse University.

With the help of an army of teaching assistants (TAs), I taught a course on compiler design for a class with over 200 students. In the days before email, it was impractical for me to help that many students individually. Instead I selected my student, Birger Andersen, to be my chief TA. Birger, who is now Associate Professor at the Copenhagen University College of

Engineering, gave me feedback from 15 regular TAs, each of whom was responsible for helping 15 students. This arrangement worked fine. Although the TAs were well-meaning, their democratic desire to be involved in all decisions was annoying to someone who had twenty years of experience in compiler design. My indirect communication with TAs, through Birger, saved me from having weekly discussions with fifteen people about my choice of textbook and philosophy of teaching.

I used my own textbook, with the modest title *Brinch Hansen on Pascal Compilers* (1985), to explain how a Pascal compiler works. Each student then used Pascal to write a complete compiler for a small programming language. The compiler project was divided into six phases, each corresponding to a chapter in the textbook. After reading a chapter, the students were ready to program the corresponding part of the project.

After one of my lectures, three female students came up to me and said: "Thanks for an interesting lecture—it happens so rarely." Before I could ask for their names, they walked away. Twenty years later, I still remember this nice compliment.

⋆ ⋆ ⋆

While I was still in Southern California, I heard that Danish universities began using Concurrent Pascal as soon as it became available from Caltech. But I didn't know it was also being used by high-tech companies, such as Brüel & Kjær, Elbau, GNT-Automatic, and ITT Standard Electric Kirk. The widespread use of Concurrent Pascal in Denmark was partly due to DIKU's requirement that every student had to solve a non-trivial programming problem for a company before graduating.

On November 6, 1985, I participated in a one-day conference on "Concurrent Pascal—Perspectives and Experience" at the Eremitage Hotel in Lyngby. The meeting was arranged by the Center for Electronics (Elektronikcentralen) and the Association for Microprocessor Electronics" (SMT). The fifty attendants represented thirty companies and research centers. I opened the conference with a talk on "Edison—the successor of Concurrent Pascal."

Risto Petersen described Elbau's use of Concurrent Pascal in dedicated microprocessor systems for farmers, dentists, and foundries.

Niels Holm Pedersen summarized Brüel & Kjærs experience using Concurrent Pascal to program electronic measurement instruments. On the morning of the conference, the newspaper Berlingske Tidende published an interview with Niels and me under the headline: "Industry and academia speak the same language."

Richard Whiffen, president of the company Enertec in Pennsylvania, gave an interesting talk about a Concurrent Pascal subset for microcomputers, called mCP. A version delivered to McDonnell Douglas in St. Louis was used to produce a digital flight controller for the F15 Eagle aircraft. The original compiler from Caltech was able to determine the memory requirement of any Concurrent Pascal program before it was executed. As Whiffen pointed out: Knowing that it was impossible to get a memory overflow in a fighter jet flying at twice the speed of sound was comforting to the pilot!

⋆ ⋆ ⋆

During a preliminary visit to Copenhagen, I had lunch with René Tang Jespersen, chairman of the GN Corporation, to discuss the possibility of consulting for him.

GN was the last remnant of the Great Northern Telegraph Company established in 1869 by the Danish Titan of industry, Carl Frederik Tietgen (1829–1901). On October 20, 1871, the Danish frigate "Tordenskiold" landed one end of a telegraph cable in Deepwater Bay, Hongkong. The other end would be landed in Shanghai. By 1894, the telegraph cables of Great Northern extended from England across Scandinavia and Russia to Japan and China. Tietgen was also a driving force behind Danish banking (Privatbanken, 1857), shipping (Det forenede Dampskibsselskab, 1866), and sugar production (De danske Sukkerfabrikker, 1872).

The telegraph was the first invention that made it possible to reach people quickly anywhere in the world. Since you paid for every word you sent, telegrams were usually brief and to the point—even in emergencies. When the Danish brig "Ane" was shipwrecked near Laguna de Terminos in the Gulf of Mexico on the morning of February 13, 1871, my great-grandfather's brother, Captain Peter Brinch, sent the following telegram (in English) to his family on the Danish island of Fanø:

ANE WRECKED NEAR LAGUNA STOP CREW SAVED

A hundred years later, war and revolution had reduced Tietgen's global empire to a handful of small, innovative companies, which included GN Batteries, GN Danavox, GN Data, GN Elmi, GN Netcom, and GN Telematic.

My meeting with Tang Jespersen took place in Great Northern's corporate headquarters on Kongens Nytorv, opposite the Royal Theater. The building was crowned by a "statue of liberty" (named Electra), who held a light globe that was turned on at night. René had worked ten years for the computer manufacturer Honeywell in Scandinavia and Belgium. After a hectic life as director of finance and administration for Honeywell Europe, he returned to Denmark as vice president of GN. When I met him in November 1983, we were both 45 years old. During our lunch, he offered me a three-year contract as an independent advisor to GN's board of directors.

In August 1984, Ernst Hede, president of GN Elmi, introduced me to one of his young engineers, Anders Raasted, who was in charge of an interesting project. They were developing a *multicomputer in a briefcase* that would be used to measure the reliability of telephone lines. The briefcase would be connected to one end of a telephone line and left alone for weeks transmitting test signals down the line. At the other end of the line, a second briefcase would receive the signals and return them to the first briefcase, which would collect data about the frequency of transmission errors. Telephone technicians would be able to inspect the measurements at any time without interrupting the real-time data collection.

Elmi asked me to design a special-purpose operating system with parallel processes for this real-time application. As I started working on the problem, the number of parallel activities soon reached a point where I found myself unable to write a clear description of what I was doing. So I asked Raasted to be patient while I designed yet another parallel programming language for real-time design.

I had already invented parallel programming languages which included monitors (Concurrent Pascal, 1975), remote procedure calls (Distributed processes, 1978), and conditional critical regions (Edison, 1981). This time I used a minimal subset of Pascal to design a secure parallel language, named *Joyce* (Brinch Hansen 1987a, 1987c). The new language was based on Hoare's idea of communicating sequential processes (CSP) which exchange messages through synchronous channels without automatic buffering. Joyce removed a major limitation of CSP by introducing *parallel recursion* in the form of processes that spawn copies of themselves.

To experiment with the new language, I developed a portable implementation of Joyce on an IBM PC (Brinch Hansen, 1987b). This was apparently the first recursive CSP language implemented on a computer. The Joyce compiler checked that parallel processes never referred to the same variables. This was essential since the multicomputer would consist of microprocessors without shared memory. The compiler also checked that every message sent from one process to another was received in a variable of the same data type as the message itself. This turned out to be one of the most frequently detected programming errors in my Joyce programs.

My next step was to use Joyce to simulate a simplified version of Elmi's real-time system. Working with Raasted's group, I defined the process structure and communication patterns of the actual real-time system. I then wrote a Joyce program with the same number of processes and communication channels. My Joyce model was, of course, greatly simplified. The circuit board that generated test signals was represented by a ten-line process that sent a sequence of integers through an output channel. Transmission errors were simulated by occasionally outputting the wrong values. Other processes simulated a real-time clock, a simple filing system that collected measurements, and a console used by technicians to inspect selected measurements.

This Joyce program was an *executable model* of the real-time system that could be studied and tested by programmers to reveal systemwide flaws, such as deadlocks, in the process structure. Today, this design method would be considered an example of "rapid prototyping."

Using my Joyce program as a model, Elmi was able to fill in the missing details and implement the final software product as a set of Pascal programs running in parallel. It turned out to be the first time Elmi had delivered a new product to its customers *ahead of schedule.*

⋆ ⋆ ⋆

Returning to Denmark was not as easy for me as I had thought. In C. P. Snow's novel "Last Things," a Jewish tycoon, Azik Schiff remarks that "coming to England as an exile, he had felt one irremovable strain: you had to think consciously about actions which, in your own country, you performed as instinctively as breathing."

I had the same experience when I lived in America. Now, many years later, I had it again in Denmark. At DIKU it was definitely not kosher for the faculty to have close ties to industry. This taboo surprised me since I had grown up in Copenhagen with a father, who had worked both in industry as a civil engineer and in academia as a professor of engineering. I remember an instance where DIKU had agreed to let IBM borrow our PCs for use in a summer course for unemployed people. My colleagues were horrified when I proposed to ask IBM in return to let us borrow one of their newest work stations.

On January 1, 1985, as the first Danish computer scientist, I was elected a Fellow of The Institute of Electrical and Electronics Engineers "For contributions to concurrent programming and operating systems." IBM graciously agreed to host the event and serve refreshments for the audience. At the award ceremony I demonstrated my Edison system for the IBM PC on a huge screen display at the IBM auditorium in Lyngby. I don't remember what my colleagues thought of this untraditional arrangement.

On another occasion, I rocked the boat as chair of a faculty committee that considered one of DIKU's PhD candidates for an assistant professorship. For obvious reasons, leading American universities rarely hire their own graduates. PhDs who have studied in the same department for years are not likely to move it in a new direction and risk offending their former advisors and their colleagues.

By the Danish rules of the game, my committee was only supposed to answer one question: Is the candidate minimally qualified to become an assistant professor? Instead, I asked my colleagues if they thought this appointment would improve the department in any way. Caught by surprise, they agreed with me that he should not be appointed. It caused a stink when the rest of the faculty heard what happened. They had all taken it for granted that he would be appointed. After that, I was viewed as someone who threatened the harmony of the department.

I regard Denmark as one of the most civilized countries on Earth with its low rates of poverty and crime, universal health care, and free education. To pay for this level of welfare, Danes pay extremely high taxes (my income taxes were 2/3 of my salary). I still consider that an acceptable price to pay for a just society.

The computer scientist, Alan Perlis, told me an amusing story about the difference between American and Danish mentality. During a visit to Regnecentralen, he asked my boss, Niels Ivar Bech, to show him a slum area in Copenhagen. So Bech drove him to a neighborhood that was poor by Danish standards. However, when they got there, Perlis said: "Niels Ivar, this is not a slum—where *are* the slums of Copenhagen?" Bech answered: "If we had any, this is where they would be!"

Like so many other foreigners, Milena and I found Americans to be some of the nicest and most hospitable people you can imagine. Shortly after we moved to Southern California, two of our neighbors, Eileen and Bob Harder, invited us over for Thanksgiving dinner. This openness towards strangers is rare in Denmark, where most people stick to their own friends.

Foreigners, who live in Denmark, have often described how difficult it is to find close friends among the Danes. When I worked for Regnecentralen, one of my colleagues was a young American, named Roger House, who married a Danish women, named Jeanne. After a couple of years, Roger returned to the United States. On his last day at Regnecentralen, he quietly asked us: "How come none of you ever invited me to your homes?" We looked at him and said: "But we didn't think you would be interested!"

After our return to Denmark, Milena and I discovered that we were no longer part of the circle of our former friends. They were happy to come to our return party—but few of them invited us back. Those who did had no interest in our stories about life in America. This feeling of alienation is the price you pay for leaving your country in search of adventure: in the end you don't belong in either country—but you have lived an exciting life. So be it!