FORTRAN – punkt zwrotny w historii programowania

Paweł Fritzkowski

Czasy, w których przyszło im żyć...

Świat, o którym będziemy rozprawiać, to świat amerykańskich korporacji i koncernów, których działalność miała nierzadko kluczowe znaczenie dla rządowych instytucji. Ale przede wszystkim mowa będzie o indywidualnościach – inżynierach i naukowcach, których dokonania wniosły wkład nie tylko w rozwój i potęgę Stanów Zjednoczonych, ale zmieniły oblicze ówczesnej nauki i inżynierii.

A były to czasy ożywionej produkcji maszyn liczących, by nie rzec: pierwszych komputerów. Bodziec rozwoju, czego nietrudno się domyślić, miał charakter militarny. W 1950 roku rozpoczął się konflikt zbrojny w Korei. Tworzenie i realizacja nowych projektów wymagały innej niż do tej pory efektywności obliczeń inżynierskich. Zainteresowanie, jak i finanse Pentagonu były ogromne, a o technologii informatycznej wykraczającej poza potrzeby armii nikt wtedy jeszcze nie myślał.

Już II wojna światowa pchnęła technikę i producentów maszyn księgujących – IBM, Remington Rand, Burroughs czy NCR – ku elektronice mającej wspomagać przemysł zbrojeniowy. Jednak wymagania rynku rosły coraz szybciej. Właściwe rozeznanie się w sytuacji przez naukowców i młodsze pokolenie menadżerów IBM usytuowało tę firmę na silnej, ogólnoświatowej pozycji w branży. W 1951 roku bowiem zaaprobowano projekt nowej maszyny obliczeniowej. Pierwszy z serii 700 komputer produkcji IBM nosił początkowo nazwę Defense Calculator, sugerującą zaangażowanie w problemy wojskowości. W rzeczywistości jednak była to maszyna ogólnego przeznaczenia, mogąca stawić czoła zagadnieniom z różnych dziedzin.

Monstrualna dla nas konstrukcja, złożona z kilku oddzielnych jednostek nieprzeciętnej wielkości, była na tamte czasy zwarta i stylowa. Niemniej jednak to szybkość była tym, co mogło imponować najbardziej. Mimo, że moc obliczeniowa komputerów lat 50-tych była tylko znikomym ułamkiem możliwości, jakie reprezentują sobą dzisiejsze – nawet te kieszonkowe – komputery.

Będąc w posiadaniu tak nowoczesnego urządzenia, firma IBM stanęła w obliczu nie napotkanego dotąd paradoksu techniki cyfrowej – ścisłej współzależności dwóch różnych sfer komputerowej branży: *software'u* i *hardware'u*. Na pytanie „jak zapewnić ciągłość pracy nowej, szybkiej maszyny?" znaleziono prostą, racjonalną odpowiedź: „należy dostarczać znacznie większą liczbę problemów do rozwiązania". Tkwił w tym sposobie jednak pewien poważny, nieznany współczesnym, szkopuł: programowanie!

Tajemna wiedza tylko dla wybranych

Opracowanie danego problemu technicznego czy naukowego w taki sposób, aby rozwiązać go za pomocą komputera, wymagało szczególnych umiejętności. Komunikacja na linii „człowiek-komputer" oznaczała bowiem dostosowanie się tego pierwszego do „niskiego poziomu" komputera.

Początkowo zrozumienie przebiegu obliczeń i nadanie mu właściwego kierunku narzucało konieczność posługiwania się wprost dwójkowym systemem liczbowym – tak nienaturalnym dla człowieka, w przeciwieństwie do przyswojonego przezeń systemu dziesiętnego, z którym wszyscy jesteśmy świetnie obeznani. Nieco później pewnym ułatwieniem okazało się użycie systemu oktalnego czy – począwszy od lat 60-tych – heksadecymalnego, które są stosunkowo łatwe do odczytania, a jednocześnie zapewniają sprawną konwersję liczb do formatu binarnego.

Ważnym krokiem ku efektywniejszemu programowaniu komputerów stało się wprowadzenie tzw. asemblerów – języków bazujących bezpośrednio na operacjach maszynowych. I tak oto kod źródłowy złożony z symbolicznych poleceń zrozumiałych dla programisty mógł być bezpośrednio przekształcony w kod maszynowy przez program tłumaczący, zwa-

ny również asemblerem. Warto wspomnieć, że twórcy i zarazem realizatorzy tej innowacyjnej idei – grupa badaczy z Cambridge University – wprowadzili nie mniej ważką dla świata programistów koncepcję procedur, a więc segmentów programu najczęściej, bo wielokrotnie, używanych.

Tak czy inaczej programowanie wydawało się być profesją tylko dla nielicznych, ambitnych i wytrwałych, operujących językiem maszynowym na tyle biegle, by zręcznie od początku do końca przeprowadzić rozumowanie w sferze rozwiązywanego zagadnienia.

A rozwiązywało się czasem miesiącami. Przedmiot badań mógł być niezwykle ciekawy, tymczasem główny ich cel przysłaniała „walka wręcz" człowieka z techniką. Po matematycznym zamodelowaniu problemu następował złożony proces maszynowej implementacji: opracowanie karty „przepływu" obliczeń w komputerze, a następnie żmudne rozplanowanie etapów obliczeń i ich zapis na kartach perforowanych – krok za krokiem, instrukcja po instrukcji. Ponadto maszyna musiała być przygotowana do każdego zestawu zadań – dziesiątki przełączników i przewodów podłączanych ręcznie do odpowiedniego obwodu.

Zaplanowanie obszernego programu mogło trwać całe tygodnie, a wykonanie obliczeń – nawet pół roku. Komputer przerywał pracę średnio co trzy minuty, oczekując na dalsze polecenia od programistów, co wymuszało na nich ciągłą obecność. Wszelkie trudności, pojawiające się w trakcie działania programu, pokonywano począwszy od odczytania i deszyfrowania liczb wyświetlanych binarnie przez tysiące żarówek, jakie posiadała maszyna. Jednak ta monotonna, męcząca rzeczywistość miała się niebawem zmienić. Nie sama, rzecz jasna.

Do odważnych świat należy

John Backus trafił do firmy IBM w roku 1950. Ale ścieżka, która go tu przywiodła, jak cała jego życiowa trajektoria zresztą, była co najmniej kręta.

Wychowany w Wilmington, Delaware, w rodzinie bogatej i prominentnej, od dziecka zafascynowany był techniką. Nie przeszkodziło mu to uczniem być raczej krnąbrnym – począwszy od prywatnej szkoły w stanie Pennsylvania, poprzez *college*, a na uniwersytecie w Virgini kończąc. Nawiasem mówiąc, na uniwersytecie John niedługo zabawił – już po dwóch semestrach został stamtąd wyrzucony.

Jak się później okazało, niezbyt udana edukacja miała niewiele wspólnego z jego rzeczywistymi możliwościami. Będąc w wojsku, uzyskał wyśmienite wyniki testów umiejętności i w rezultacie wysłano go na uniwersytet w Pittsburgu, a potem do Haverford College. Uczęszczał też do Medical College na Manhattanie, lecz medycyna go nudziła. Dziwnym zbiegiem okoliczności interesująca dla niego stała się matematyka. I tak oto rozpoczął studia w Columbia University, otrzymując zdecydowanie wszechstronną wiedzę i zdobywając dyplom z matematyki w roku 1950.

To właśnie w tym czasie Backus udał się do siedziby IBM przy Madison Avenue, zaciekawiony potężną maszyną obliczeniową, o której było wtedy głośno. Jednak nie był to jeszcze ten cudowny IBM 701, którym przyszło zachwycać się kilka lat później. Jego poprzednik, ukończony w 1948 roku komputer – prawdopodobnie najpotężniejsza wtedy maszyna – określany był skrótem SSEC (Selective Sequence Electronic Calculator) lub częściej jako Super Calculator. Backus wspomniał kobiecie, która go oprowadzała, że właśnie kończy studia, jest matematykiem, a już znalazł się u współtwórcy SSEC, Roberta Seebera. Po serii pytań, które nazwać by można „łamigłówkami", Seeber zatrudnił go natychmiast – jako programistę.

A był to okres przejściowy nie tylko dla IBM, ale także dla całego przemysłu i technologii komputerów. Był to ów czas zmian zapoczątkowanych przez II Wojnę Światową. Super Calculator odzwierciedlał innowacyjne trendy. Był przede wszystkim projektem naukowym, wykraczającym poza ustalone dotychczas granice, pozwalającym zdobywać nowe doświadczenia.

Z możliwości tych korzystać miał właśnie John Backus. I faktycznie pracował nad potężnymi programami obliczeniowymi, pozwalającymi na przykład wyznaczać położenie księżyca czy planet w dowolnym czasie na przestrzeni lat. Jak sam kiedyś powiedział: „To była czysta nauka".

Jednak ówczesne programowanie, przypomnijmy, było nawet dla biegłego badacza raczej udręką. Stojący w obliczu powszechnych wtedy trudności Backus z takim stanem rzeczy pogodzić się nie zamierzał. Wręcz przeciwnie – podjąć chciał działania, aby rzecz całą uprościć. „Doszedłem do wniosku, że musi istnieć lepsza droga. Po prostu trzeba było ułatwić ludziom programowanie".

Na szczęście plany te nie pozostały tylko w sferze ulotnych myśli, lecz okazały się zalążkiem brawurowego przedsięwzięcia – praktycznego poszukiwania owych lepszych dróg. Pod koniec roku 1953 John Backus w krótkim liście do swego szefa przedstawił propozycję nowego, programistycznego projektu. Zwrócił w nim szczególną uwagę na ekonomiczne aspekty problemu. Przede wszystkim przestawianie maszyny na potrzeby nowego zadania zazwyczaj wymagało obecności trzydziestu programistów; ze względu na ich wynagrodzenia kosztowało to co najmniej tyle, ile cały komputer. Ponadto kodowanie oraz szukanie i usuwanie usterek stanowiło

trzy czwarte kosztów związanych z obsługą maszyny. A przecież stopniowo jakość sprzętu rosła, cena malała, tak więc wszystko wskazywało na to, iż proporcje te zmienią się na niekorzyść procesu programowania. Przełożony Backusa, Cuthbert Hurd, ten naglący problem doskonale rozumiał, toteż wysunięty plan badawczy natychmiast zaaprobował.

Zespół nieustraszonych

Pierwszym sprzymierzeńcem Johna stał się Irving Ziller. Absolwent Brooklyn College trafił do firmy IBM w roku 1952. Jego zadaniem było programowanie paneli sterowania (ang. *plug-boards*) ówczesnych kalkulatorów, polegające na podłączaniu do owych płyt gąszczu przewodów. Ziller szybko odnalazł się w tej trudnej pracy, ale wydawała mu się ona na tyle mało ekscytująca, że Backus bez kłopotu pozyskał go dla swojego projektu.

Krótko potem dołączył do nich Harlan Herrick, matematyk z Iowa State University i znakomity szachista. Mimo otrzymania stypendium na Yale University, zrezygnował i po przeczytaniu artykułu o maszynie SSEC postarał się o pracę w IBM. W firmie uważany był za niezwykle utalentowanego programistę. W momencie przystąpienia do zespołu Backusa miał za sobą pięcioletnie doświadczenie zarówno z maszyną SSEC, jak i komputerem 701. Przesiąknięty stylem programowania tamtych czasów Herrick był najbardziej sceptyczny z całej trójki. Nie wierzył, że człowieka-programistę można zastąpić językiem – takim, który wygeneruje kod maszynowy z efektywnością zbliżoną chociażby do efektywności uzyskiwanej konwencjonalnymi sposobami.

Jego podejście nie było jednak odosobnione. Spośród ludzi należących do przemysłowego *establishmentu* chyba nikt nie wierzył w powodzenie projektu. Wysoko stawiający sobie poprzeczkę zespół był jakby odsunięty w cień – ingerencja kierownictwa była nieznaczna, a cała działalność – raczej nieoficjalna. Na przykład Backus nigdy nie ustanowił formalnego budżetu, choć z czasem projekt się rozrastał, a termin jego ukończenia coraz bardziej się oddalał.

Powtórzmy raz jeszcze: celem zespołu było zautomatyzowanie ręcznie przeprowadzanego procesu programowania. Przy czym, w zamyśle Johna Bachusa, chodziło tu o wykroczenie daleko ponad poziom asemblera i przełamanie istniejącej konwencji „jeden do jednego": jedna linia kodu źródłowego miała być tłumaczona na szereg instrukcji kodu maszynowego. Realizacja tak ambitnego planu przyniosłaby nie tylko ogromny postęp technologiczny, ale także pewne przemiany kulturowe środowiska programistycznego.

Należy wyraźnie podkreślić, iż stworzenie języka programowania było niejako celem pośrednim. Backus, Ziller i Herrick rozumieli, że sukces zależeć będzie o wiele bardziej od skuteczności działania kompilatora niż od języka samego w sobie. „Po prostu tworzyliśmy język w miarę, jak posuwaliśmy się naprzód. – tłumaczył Backus – Konstrukcji języka nie traktowaliśmy jako zagadnienie trudne, a tylko jako łatwy wstęp do rzeczywistego problemu: zaprojektowania kompilatora, dającego wydajne programy".

Innych nieudane próby

Byli tez inni, którym przyświecał ten sam cel. Jedną z osób najbardziej otwarcie nawołujących do zmiany była Grace Hopper, pracująca w latach 50-tych dla Remington Rand. Ów powszechny programistyczny problem jawił jej się w podobnych barwach, co Johnowi Backusowi. Widziała potrzebę stworzenia pewnego rodzaju interfejsu zorientowanego na człowieka i użycia komputera do wygenerowania kodu maszynowego. Wizję taką – nazywała ją „automatycznym programowaniem" – prezentowała często na różnego rodzaju spotkaniach poświęconych światu komputerów. Dla maszyny UNIVAC (Universal Automatic Computer) napisała system spajający fragmenty kodu w jeden program i nazwała go kompilatorem A-0 (późniejsze wersje: A-1 oraz A-2). W rzeczywistości był to raczej zbiór pewnych pomocy programistycznych aniżeli kompilator w dzisiejszym tego słowa znaczeniu.

Pierwszy, który mógłby sprostać owej współczesnej definicji, powstał najprawdopodobniej w ramach rządowego projektu realizowanego w Massachusetts Institute of Technology (MIT). W roku 1954 tamtejsi uczeni, J. Halcombe Laning i Neal Zierler, napisali program zamieniający równania algebraiczne na kod maszynowy. Podczas wizyty w MIT pracy kompilatora przyjrzeli się Backus i Ziller, a można ją podsumować słowami tego drugiego: „To była bardzo dobra robota, koncepcyjnie bardzo dobrze wykonana. Ale zastosowali podejście akademickie. Nie mogli zadbać mniej o efektywność". W istocie wykonanie programu skompilowanego nowatorską metodą trwało około 10 minut dłużej niż tego otrzymanego starym sposobem.

Pierwsze wyjście na świat

O wiele skuteczniej z wielkim wyzwaniem lat 50-tych radziła sobie grupa z IBM, a efektem jej wysiłków stał się oczywiście FORTRAN. Nazwę tę zaproponował w 1954 roku sam Backus, a miała ona precyzyjnie określać owoc realizowanego projektu, czyli zarówno język, jak i kompilator. I faktycznie FORTRAN to po prostu skrót od słów: *FORmula TRANs-*

lator, znaczących mniej więcej tyle, co „system tłumaczący wzory".

10 listopada 1954 roku światło dzienne ujrzał napisany przez zespół tekst zatytułowany: „*Preliminary Report: Specifications for the IBM Mathematical FORmula TRANslating System*" („Raport Wstępny: Dokumentacja Matematycznego Systemu FORTRAN firmy IBM"). Przede wszystkim zaprezentowano tu matematyczny żargon zagnieżdżony niejako w FORTRANie, a także zasady posługiwania się podstawowymi elementami składni języka, jak pętla DO, instrukcja warunkowa IF, czy polecenia GOTO oraz STOP. Co więcej, raport – pisany jakby w duchu nowoczesnego marketingu – wyrażał plany Johna Backusa i jego optymistyczną wizję co do przyszłości FORTRANa. Zawierał też, przytaczane już wcześniej, ekonomiczne argumenty przemawiające za zautomatyzowaniem procesu kodowania.

Chyba największą rolę w tamtych czasach odgrywało potencjalne otwarcie świata komputerów dla szerszej rzeszy ludzi – owo uprzystępnienie programowania, które przecież było przedmiotem rozważań różnej maści wizjonerów. Jednak teraz rzecz ta miała miejsce naprawdę. Jak głosił raport, programowanie przy użyciu FORTRANa wymaga wiedzy o wiele mniej rozległej niż ta niezbędna do bezpośredniego kodowania. „W istocie znacząca ilość informacji, które programista musi posiadać o systemie FORTRAN, jest już zawarta w jego znajomości matematyki".

Jak by się mogło wydawać, dzieło badaczy z IBM było milowym krokiem, a nawet rewolucją w dziedzinie komputerów i jako takie powinno ówczesnych ludzi zachwycić. Ale pierwsze próby wprowadzania zmian, początki niesienia nowej nauki o programowaniu okazały się niezwykle trudne. Z dokumentacją w ręku Backus, Ziller i Herrick wyruszyli na wędrówkę po kraju – na spotkanie z użytkownikami maszyny 704, dla której innowacyjny system, pamiętajmy, pierwotnie był przeznaczony. Odwiedzając Waszyngton, Los Angeles, Pittsburgh czy Albuquerque, głosili własną wizję przyszłości programowania i roli FORTRANa. Liczyli jednocześnie, że uda im się wzbudzić zainteresowanie projektem i uzyskać pewne użyteczne sugestie od praktyków.

Jakkolwiek nadzieje te wydawały się płonne. Dla ludzi działających „po staremu" raport zespołu jawił się jako myślenie życzeniowe, kojarzące się zapewne ze złudnymi ideami „automatycznego programowania" autorstwa Grace Hopper i innych. Realistycznie patrzący odbiorcy zwyczajnie nie wierzyli w skuteczne działanie języka i kompilatora. FORTRAN wydawał się wykraczać daleko poza horyzont realności – zbyt daleko, by w jego siłę nie wątpić.

Nowi ludzie

Mało satysfakcjonująca podróż po kraju wniosła w dalsze prace zespołu sporo zapału i motywacji. Przekonani o właściwym kierunku projektu, jak i jego dobrej realizacji, twórcy FORTRANa czuli potrzebę pokazania światu, że ich nowatorskie idee nie były naiwnymi mrzonkami. Na tę chwilę przyszło im jednak jeszcze trochę poczekać. Chociaż termin zakończenia prac nie wydawał im się zbyt odległy. „Naprawdę zawsze czuliśmy, że skończymy to w sześć miesięcy od chwili obecnej. – mówił Backus – Ale zrobiły się z tego prawie trzy lata".

W roku 1955 Backus zaczął poszukiwać nowych osób, które mogłyby przyłączyć się do *FORTRAN teamu* i wspomóc go swoją wiedzą czy doświadczeniem. Rozglądał się za ludźmi nieprzeciętnymi, zaczął więc od MIT – jednego z głównych centrów obliczeniowych, zresztą blisko związanych z IBM. Przysłano stamtąd jednego z najlepszych programistów, a był nim Sheldon Best. Z pomocą przyszła też korporacja United Aircraft i w efekcie do zespołu dołączył Roy Nutt. To była prawdziwa zdobycz – niezwykły programista potrafiący „wykonać" program w głowie, niczym maszyna, a następnie napisać kod pozbawiony jakichkolwiek błędów.

Ale również wewnątrz firmy John Backus znajdował bystrych kandydatów, choć tu bywało gorzej z doświadczeniem. Robert Nelson, były kryptograf, zatrudniony od niedawna w IBM, przepisywał na maszynie teksty naukowe. Lecz Backus zauważył u niego talent techniczny. I faktycznie Nelson szybko stał się znakomitym programistą, decydującym o powodzeniu projektu.

Richard Goldberg, doktor matematyki z New York University, niegdyś nauczyciel w *college'u*, również niewiele wiedział o programowaniu. Ale z bardzo dobrymi wynikami ukończył trzymiesięczny kurs i dołączył do grupy.

Natomiast Lois Habt trafiła do IBM zaraz po ukończeniu Vassar College. Jako świetna studentka, obeznana z naukami ścisłymi, pracowała w Bell Labs – jednym z najznamienitszych ośrodków badań i rozwoju. Jakkolwiek absolwentkę przyciągnęła właśnie firma IBM. „Powiedzieli mi, że to będzie praca polegająca na programowaniu komputerów. Miałam tylko mgliste pojęcie o tym, co to jest". Habt okazała się jednak niezwykle biegła w tej materii, więc przydzielono ją do *FORTRAN teamu*.

David Sayre z kolei był krystalografem prowadzącym badania naukowe w dziedzinie biofizyki w University of Pennsylvania. W działalności tej często stosował obliczenia komputerowe, ale – zawiedziony możliwościami maszyny uniwersyteckiej – zmuszony był korzystać w końcu z IBM 701. Samo programowanie wysoce go satysfakcjonowało. „Kie-

dy uruchamiałeś program, działo się coś oczekiwanego, a jeśli nie, to istniała logiczna tego przyczyna". Gdy Sayre dołączył do IBM, nie tylko pisał aplikacje naukowe, ale zagłębił się również w czyste programowanie. To dlatego Backus zwerbował go dla celów realizowanego projektu.

I tak oto wzbogacony o znakomitych specjalistów z różnych środowisk zespół mógł teraz kontynuować pracę na rzecz doskonalenia języka FORTRAN. Te wzmożone wysiłki, widziane nawet z dzisiejszej perspektywy, nie poszły na marne – wręcz przeciwnie: dały rezultat w postaci imponującego osiągnięcia w dziedzinie informatyki.

Po owocach ich poznacie

Podstawowym wyzwaniem podejmowanym zawsze w procesie tworzenia kompilatora jest uchwycenie problemu – widzianego i formułowanego przez ludzi – tak by mógł być on rozpracowany przez maszynę. To właśnie sposób, w jaki John Backus zdecydował się kwestię tę rozwiązać – oryginalny podział działań kompilatora na etapy – wniósł do fenomenu FORTRANa chyba najwięcej.

Praca kompilatora została bowiem rozłożona, można powiedzieć, na czynniki pierwsze według zadań operacyjnych. Na początku system dokonywał skanowania kodu źródłowego, czyli wstępnej interpretacji słów kluczowych, symboli algebraicznych i angielskich skrótów. Dopiero teraz następowała dogłębna analiza programu, szczególnie koncentrująca się na najczęściej powtarzanych w nim operacjach. Trzeci etap to właściwe zorganizowanie zebranych już instrukcji, aby ich wykonanie trwało możliwie najkrócej. I w końcu finalizacja, a więc skompilowany program należało przekształcić w kod maszynowy.

Czynności te kompilator fortranowski wykonywał z właściwą sobie elegancją, by nie powiedzieć – inteligencją. Zespół z IBM stworzył po prostu dobrze zaprojektowane, a zarazem bardzo złożone oprogramowanie, funkcjonujące według obmyślonych zasad i realizujące założone cele. Ale rezultat zdumiewał czasem nawet samych twórców. Kompilator, jak stwierdził Backus, dokonywał „zadziwiających transformacji" – modyfikował wprowadzone wyrażenia i kolejność obliczeń. Co więcej, prześledzenie tych zmian ujawniało niezwykłą efektywność tych działań – kompilator zdobywał się na kroki, o których „my sami jako programiści nie pomyślelibyśmy, aby je podjąć".

Mimo że FORTRAN jako całość był efektem niepospolitego połączenia dwóch sposobów rozumowania – koncepcyjnego i proceduralnego – to właśnie niskopoziomowe zmagania były najbardziej mozolne. Tym bardziej, że uzyskanie dużej efektywności „nowych dróg" programowania w powszechnym mniemaniu było poza zasięgiem kogokolwiek. Programming Research Group – bo tak pierwotnie nazywano zespół działający pod przewodnictwem Backusa – była więc, jako się rzekło, grupą pozostającą nieco z boku, próbującą dokonać czegoś, czego jeszcze wcześniej nie dokonano. Ale tworzyli ją przecież młodzi zapaleńcy – ludzie inteligentni, blisko ze sobą związani, przepełnieni optymizmem i energią.

Czas maszynowy był zasobem raczej ograniczonym, tak więc kompilator FORTRANa uruchamiany był głównie w nocy. I choć grupa Johna Backusa wypracowywała innowacyjne metody programowania, to środki, którymi się posługiwała, były jak najbardziej tradycyjne: karty perforowane, mające stopniowo wychodzić z użycia dopiero z początkiem lat 60-tych. Dlatego kod pisano najpierw na kartce, dopiero później przenoszono go na ów prymitywny nośnik.

Organizacja pracy wynikała bezpośrednio z zadań realizowanych przez kompilator. Zespół podzielony został na jedno-, dwu- lub trzyosobowe komórki – autonomiczne jednostki, wypełniające powierzone zadania przy użyciu dowolnych – możliwie najefektywniejszych – technik, jednak kompatybilnych z tymi stosowanymi przez sąsiednie sekcje programistyczne. Lois Haibt, odpowiedzialna za analizę przepływu – przede wszystkim wychwytywanie potencjalnych przeciążeń kompilatora – tak opisała współpracę wewnątrz grupy: „To była taka atmosfera, w której, jeśli nie widziałeś błędu w swoim programie, mogłeś zwrócić się do osoby obok. (...) Po prostu uczyliśmy się wszyscy razem".

Do Zillera i Nelsona należało chyba najtrudniejsze zadanie: analiza i optymalizacja tzw. wewnętrznych pętli kompilatora. Badali najczęściej przeprowadzane operacje i starali się znaleźć najwydajniejszą procedurę ich wykonania. Metodą prób i błędów udoskonalali swe rozwiązania, wykluczając coraz więcej instrukcji z owych wewnętrznych pętli, minimalizując tym samym czas obliczeniowy.

Pierwszy język wysokiego poziomu

Ostateczny rezultat kilkuletnich działań zaprezentowano w lutym 1957 roku na Western Joint Computer Conference w Los Angeles. Uczestnikami spotkania byli członkowie grupy SHARE, którą stanowiły firmy wykorzystujące komputery IBM. Demonstracja możliwości nowego języka polegała na rozwiązaniu realnych problemów obliczeniowych na dwa sposoby: za pomocą programów stworzonych przy użyciu asemblera oraz tych napisanych w FORTRANie. Nie trudno się domyślić, iż na przekór sceptykom te ostatnie okazały się pod względem efektywności dorównywać programom tradycyjnym.

Nie bez powodu owa szybkość działania jest tak często akcentowana. Bez tego atutu FORTRAN mógłby nie zostać przez świat obliczeń zaadoptowany. Duża wydajność „nowych dróg" programowania była po prostu z ekonomicznego punktu widzenia praktyczna.

Drugim wielkim osiągnięciem *FORTRAN teamu* był język sam w sobie. Jako swoista mieszanka angielskich słów i algebry znakomicie odzwierciedlał matematyczne aspekty rozwiązywanego problemu. I znów pragmatycznie rzecz ujmując, podniesiono poziom komunikacji z komputerem – bliżej człowieka. Właśnie dlatego FORTRAN jest nazywany pierwszym językiem wyższego poziomu.

Podwójny sukces grupy z IBM uczynił FORTRANa językiem niezwykle żywotnym. I choć dla większości młodych informatyków i programistów należy on tylko do historii, współcześnie co kilka lat wprowadzane są nowe, zmodernizowane standardy języka. Jednak mała popularność FORTRANa dziś nie przekreśla jego dużej przydatności do projektów trudnych i zaawansowanych. Ze względu na długą przeszłość bowiem FORTRAN zorientowany jest głównie na programowanie obliczeniowe. Dlatego stosowany jest przede wszystkim do celów naukowych, wymagających użycia wyrafinowanych metod numerycznych.

Mimo że informatyka współcześnie zmierza w innych kierunkach i stawia sobie coraz to nowe cele, rolę FORTRANa w jej historii trudno przecenić. Stanowi on ów „milowy krok", przez wielu nazywany „punktem zwrotnym". Wszak FORTRAN sprowadził świat programistyczny na drogę w błyskawicznym tempie wiodącą do narzędzi nowej generacji.

Tekst powstał na podstawie artykułu *FORTRAN The Early Turning Point* opublikowanego przez portal About.com
http://inventors.about.com/od/fstartinventions/a/Fortran.htm