

A

Association pour un conservatoire de l'informatique et de la télématique

10 bis, rue Ampère - 38000 - Grenoble - FRANCE

Tel : +33 (0)4 76 48 43 60

Web : www.aconit.org Courriel : info@aconit.org

C

Bulletin No 17 : décembre 2003

O

Sommaire

Mot du Président2

Les nouvelles de l'association3

Un dossier spécial : le cinquantenaire de Fortran

Quel avenir pourrait on prédire à un langage

« vieux » de 50 ans, à côté de Java, C++, C# ?5

Cinquante ans après, ils racontent :

Un message de John Backus.....6

(traduction).....7

Fortran à tout faire8

Downward to (la descente aux enfers vers) Fortran11

Une curiosité : la variation des constantes12

Recevoir une copie électronique du bulletin ?.....12

N

I

T

décembre

2003

Ils nous soutiennent :



Mot du président

Voici deux mois écoulés depuis le retrait de notre président.

Vos deux vice-présidents ont été amenés depuis à multiplier les contacts extérieurs. Nous avons en ce moment deux grandes opérations en cours :

- Le déménagement : la Ville de Grenoble désire reprendre les locaux que nous occupons actuellement dans les bâtiments Cémoi, rue Ampère. Dans le cadre d'une convention Ville de Grenoble - Métro (agglomération) et peut-être Conseil Général, nous pourrions disposer temporairement de locaux à Fontaine, plus propres, plus pratiques, plus sains, mais plus petits...

Inutile de dire que le déménagement, l'organisation des rangements sont des opérations difficiles et à grand risque pour la collection, mais si c'est l'occasion d'établir une première convention avec les autorités de tutelle, c'est aussi une chance pour l'association.

- Nous avons entrepris la rédaction d'un « Projet Scientifique et Culturel », que nous désirons établir sous le contrôle d'un « Comité Scientifique / Comité de pilotage » en cours de composition. Ceci devrait nous fournir une base officielle pour discuter d'une convention avec les différentes autorités de tutelle.

En effet, nous devons impérativement dépasser le stade associatif simple : la taille de notre collection, l'ensemble des projets de recherche, conservation, animation scientifique qui gravite autour, dépassent aujourd'hui les possibilités d'une simple association. Nous souhaitons parvenir rapidement à un statut associant privé-public (Établissement Public de Coopération Culturelle, fondation...) qui permettrait de pérenniser notre action et nos emplois.

*Les vice-Présidents
Ph. Denoyelle
J. Pain*



La mezzanine et les bureaux

Locaux Isnard à Fontaine

L'espace de stockage



Les nouvelles de l'association

Ces mois de novembre et décembre, Aconit a subi quelques changements positifs. Nous avons accueilli Agnès Félard parmi nous pour une durée d'un mois afin d'aider Cécile Hamadou dans la prise de ses fonctions. En tout premier lieu, l'association souhaite la bienvenue à ses trois nouveaux adhérents (Matthieu Boujonnier, Stéphane Pérez et Jean-Luc Wiczorek).

Matthieu Boujonnier et Stéphane Pérez se sont déjà attelés à la tâche. En effet, ils travaillent sur un projet particulièrement intéressant qui porte sur les machines ATARI. Ils se sont attaqués à la remise en route de deux machines datant de la période de 1978 à 1984. Un forum va être, prochainement, créé sur notre site ce qui permettra à tous de connaître la suite des événements.

Nous avons eu confirmation d'un futur (proche) déménagement dans les anciens bâtiments d'Isnard à Fontaine.

La visite des locaux s'est déroulée le mercredi 3 décembre en présence de Dominique Rillh pour la mairie de Grenoble, Véronique Votta de la MÉTRO, Philippe Denoyelle, Flore Gully, Jacques Pain, Pierre Thorel, Hans Pufal et Bernard Troulet pour ACONIT. Jacques Pain est chargé de la préparation du déménagement : inventaire « express » et coordination avec les déménageurs. Les négociations avec la mairie se poursuivent à l'heure actuelle.

Les locaux proposés sont composés

de deux parties :

- un hangar d'environ 684 m² ;
- des bureaux (en mezzanine) d'environ 100 m².

Ces locaux présentent un certain nombre d'avantages :

- une bonne accessibilité (rampe d'accès, entrée garage pour les machines) ;
- un espace vitré pouvant servir de lieu d'exposition ;
- des bureaux agréables (mais très divisés : il faudra fractionner la bibliothèque) ;
- espace de stockage propre et chauffé, mais de surface nettement réduite : une organisation stricte est à prévoir.

D'autre part, la réalisation du Projet Scientifique et Culturel d'Aconit est en cours.

Elle se déroulera en trois étapes :

- formulation du bilan de l'existant : analyse de la collection (restauration, acquisition), des activités (expositions, visites, conférences, etc.) et de l'administratif (employés, bureau, conseil d'administration, budget) depuis trois ans ;
- formulation des principales orientations du projet ;
- formulation du projet scientifique et culturel.

Une commission composée de professionnels validera les étapes 2 et 3. Le bilan de l'existant est déjà rédigé. Flore Gully contacte actuellement les futurs membres de la commission de validation.

Les différents projets soutenus par nos partenaires culturels progressent bien :

Mallette pédagogique

Les grandes lignes ont été précisées et il fut décidé de mettre en avant l'esprit de découverte de l'enfant. Nous devons rencontrer prochainement la Délégation Académique de l'Action Culturelle.

Exposition « Informatique à la Une »

Une réunion de mise au point a eu lieu avec le CCSTI le 02/12/2003. Une rencontre avec la scénographe de l'exposition a été fixée pour passer à l'étape de réalisation.

Exposition « Art, photo, informatique »

Les contacts sont en cours avec Fluid Images et le CCSTI.

Du point de vue événementiel...

Maurice Geynet a donné, le jeudi 4 décembre, une *conférence au laboratoire de Physique Subatomique et de Cosmologie (LPSC)*, autour de l'histoire des mémoires informatiques, suivie par la projection du diaporama « Mémoires d'hier et d'aujourd'hui » réalisé avec Hans Pufal pour les conférences de l'IMST 2003. Son exposé a eu un franc succès puisqu'un certain nombre de physiciens qui n'avaient pas pu y assister lui ont réclamé une autre conférence au mois de février.

Aconit a participé aux différentes rencontres proposées par la *Conservation du Patrimoine en Isère* ces derniers mois.

Nous avons participé à l'*Assemblée Générale de l'AHTI*, du 10 décembre,

qui était précédée par le colloque sur l'histoire de la micro-informatique en France et en Europe, au cours duquel le premier micro-ordinateur, le MICRAL N a été remis au Conservatoire des Arts et Métiers par son concepteur François Gernelle en présence de Jean-François Cervel Directeur adjoint du cabinet de Claudie Haigneré.

A cette occasion Jacques Pain, au nom des associations AHTI, FEB et ACONIT a pu recueillir l'assurance du soutien du ministère dans notre projet de Conservatoire. Ceci en présence de Christian Scherer chargé de mission à la direction de la technologie du ministère de la recherche et de Daniel Thoulouze directeur du musée du CNAM.

Concrètement un axe de travail en coopération avec le CNAM a été fortement recommandé par Jean-François Cervel auquel il assurerait son appui.

Nous avons accueilli dans nos locaux, le jeudi 11 décembre, l'*assemblée générale du club des utilisateurs IBM Rhône Alpes*. Une visite de la collection ainsi qu'une présentation des projets de restauration a été organisée, l'après-midi par Jacques Pain et Hans Pufal.

Décembre 2003

Future exposition ?



50
ANS

Un dossier spécial :
« Le cinquantième de “FORTRAN” »

Quel avenir pourrait on prédire à un langage
« vieux » de 50 ans, à côté de Java, C++, C# ?

Sir Charles Anthony Richard Hoare ou Tony Hoare, disait en 1982 :

“I don't know what the language of the year 2000 will look like, but it will be called Fortran”

Ce qui voulait dire : « je ne sais pas à quoi ressemblera le langage de l'an 2000, mais il s'appellera Fortran ».

FORTRAN I, II, III, IV, 66, 77, 80x, 90, 95, HPF, F, 2000...

Une évolution constante du langage, qui a inspiré beaucoup d'autres langages, mais également qui s'est inspiré de ces mêmes langages, pour évoluer !

Sa gestation de près de 3 ans, dans ou sur un IBM 701 sera accomplie par toute une équipe, dirigée par John Backus.

Dès sa naissance : IBM Mathematical Formula Translation System, familièrement FORTRAN ou Fortran, obtient un succès auprès de tous les scientifiques, grâce à une bibliothèque de formules mathématiques fournie dès l'annonce du langage sur le 704.

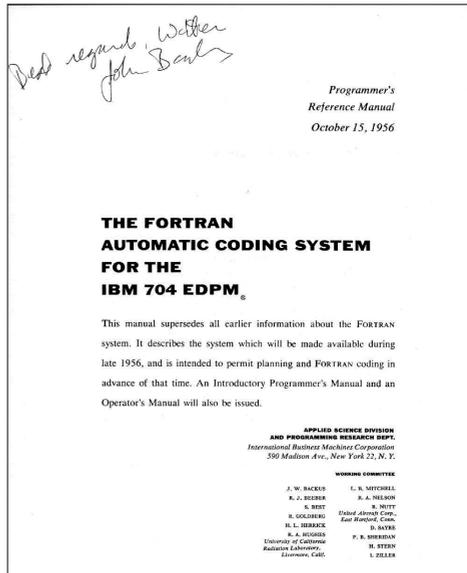
Sa facilité d'écriture et surtout sa rapidité d'exécution, le font devenir le langage favori des chercheurs, mathématiciens, physiciens, chimistes, biologistes, géographes, etc., et rapidement de tous les ingénieurs désirant exécuter

des calculs de structure, d'effort, de balistique, d'électronique, électrique, etc.

Il est surprenant d'apprendre qu'il n'est vraiment pas rare de découvrir que certains programmes ou certaines parties de programme – qui ont été écrits il y a plus de 30 ans – tournent encore aujourd'hui.

Le plus extraordinaire, c'est que certains programmes écrits il y a 30 ans trouvent seulement leur utilité aujourd'hui, dans les nano-technologies !

Jacques Pain



50 ans après, ils racontent... ..

Un message de John Backus

1) *What, if any, previous work did you build on in the early development phases?*

None that I can think of. The “compilers” that preceded Fortran were extremely primitive and had source languages that were very machine-like.

2) *Was your initial FORTRAN proposal in any way the result of user demand?*

Not at all.

3) *What was the first machine you used for FORTRAN developments?*

The IBM 704.

4) *Did you participate in the evolution of FORTRAN : 77, 80x, 90, 95, HPF F, 2000, and if so in what way?*

No.

5) *What languages do you think were inspired by the design of FORTRAN?*

I’m not sure, but it appears that Sperry-Rand’s Math-Matic and Perlis’ IT were inspired by the Preliminary Report of the Fortran project (11/10/1954). They got under way shortly after it appeared. Later, many “high level languages” adopted many of Fortran’s features, with many variations.

6) *How would you like to see FORTRAN 2000 develop?*

No opinion.

7) *Can you say a few words about your career before FORTRAN and how it led to the FORTRAN proposal.*

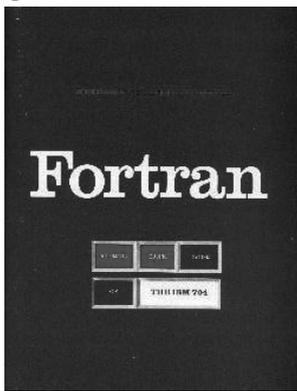
I was a programmer for the IBM SSEC computer and the IBM 701 before I started work on Fortran. I headed a small project to develop an interpretive system for the 701 called SPEEDCODING, which provided three-address floating point operations with indexing on the single-address fixed point 701, which did not have indexing capabilities. Although SPEEDCODING was a lousy system, it was used by most of the 18 701 installations rather heavily.

8) *Do any copies of the very early work still exist in any form (listings, card decks, tapes)?*

Some people I know are trying to track down a copy of the original Fortran compiler. Irv Ziller heard that the Smithsonian Institution is collecting such things and may have a copy. A friend of mine is pursuing it. I’ll let you know what comes up. There are a few copies of the original manuals still around. I sent all the old stuff I had to the US Library of Congress.

I hope the above is helpful. Good luck with your project.

John Backus, Sc.D.



Un message de John Backus (Traduction)

1) *Sur quel travail précédent (s'il y a lieu) vous êtes vous appuyé dans les premières phases du développement ?*

Sur rien dont je puisse me rappeler. Les « compilateurs » qui précédaient Fortran étaient extrêmement primitifs et avaient des langages sources qui étaient proches de la machine.

2) *Est-ce que votre proposition Fortran résultait d'une quelconque façon d'une demande des utilisateurs ?*

Pas du tout.

3) *Quelle était la première machine que vous avez utilisée pour les développements du Fortran ?*

L'IBM 704.

4) *Avez vous participé à l'évolution du Fortran : 77, 80x, 90, 95, HPF F, 2000, et si oui, de quelle façon ?*

Non.

5) *À votre avis quels langages se sont inspirés de l'étude du Fortran ?*

Je ne suis pas sûr, mais il semble que Sperry-Rand's Math-Matic and Perlis'IT se sont inspirés du Rapport Préliminaire du projet Fortran (10/11/1954). Ils ont émergé peu après l'apparition de Fortran. Plus tard, de nombreux « langages évolués » ont adopté de nombreuses fonctions de Fortran, avec beaucoup de variations.

6) *Comment voudriez vous voir se développer FORTRAN 2000 ?*

Pas d'opinion.

7) *Pouvez vous dire quelques mots sur votre carrière avant Fortran et comment cela vous a conduit au projet Fortran ?*

J'étais programmeur pour le calculateur IBM SSEC et l'IBM 701 avant de commencer à travailler sur Fortran. Je conduisais un petit projet pour développer un système d'interprétation pour le 701 appelé SPEEDCODING, qui fournissait des opérations arithmétiques flottantes (trois adresses) avec indexation sur l'opération arithmétique fixe (une adresse) du 701, qui n'avait aucune capacité d'indexation.

Bien que SPEEDCODING soit un système mal foutu, il était utilisé plutôt intensivement sur la plupart des 18 installations de 701.

8) *Est-ce qu'il existe encore des exemplaires du tout premier travail sous quelque forme que ce soit (listages, paquets de cartes, rubans) ?*

Quelques personnes que je connais essaient de pister une copie du compilateur Fortran original. Irv Ziller a appris que la Smithsonian Institution collecte ce genre de choses et pourrait avoir un exemplaire. Un de mes amis est à sa recherche. Je vous ferai savoir si ça marche. Il reste quelques exemplaires des notices d'origine. J'ai envoyé tout le paquet que j'avais à la Bibliothèque du Congrès des États Unis.

J'espère que ceci vous sera utile. Bonne chance avec votre projet.

John Backus, Sc.D.

Fortran à tout faire...

Je n'ai certes pas l'intention de me présenter comme un pionnier du Fortran : je ne l'ai découvert qu'en 1966. Mais dès le début ce fut pour moi un « outil » efficace que j'ai mis à toutes les sauces...

Plantons le décor

En 1966, je travaillais à la CAE sur la francisation de matériels TRW pour la marine française ⁽¹⁾. Je disposais alors d'un CAE 130 à la phénoménale vitesse horloge de 333 kHz, doté en version de base d'une mémoire à tores de 8192 mots de 15 bits. J'étudiais les périphériques, ce qui m'a très vite amené à développer de petits programmes de tests en langage machine. Ce n'était pas ça qui allait remplacer ma règle à calcul pour les multiples petits calculs du métier !

C'est alors qu'un ami programmeur m'a mis entre les mains la notice du FORTRAN II et la galette de ruban perforé... Ce fut une passion immédiate ! Quelques heures après, ayant survolé les 80 (?) pages de la notice, je testais mon premier programme.

Le processus de compilation

Je ne sais plus aujourd'hui si le compilateur fonctionnait avec un seul bloc mémoire. Je suis par contre sûr qu'il tournait avec deux (16384 mots). Sauveriez vous le refaire aujourd'hui ?

Il n'était évidemment pas question de stocker les informations intermédiaires en mémoire. Les périphériques standards étant une machine à écrire IBM Selectric à boule comme clavier/imprimante,

un lecteur et un perforateur de rubans, tout devait être organisé autour...

Voici donc le processus pas à pas :

Compilation

- Monter et charger le 1^{er} ruban perforé (1^{ère} passe – environ 100 m). Déposer le ruban.
- Monter le ruban perforé source. Relancer le programme.
- Le ruban est lu ligne par ligne ; chaque ligne source est imprimée sur la machine à écrire, avec les messages d'erreur ; un code intermédiaire est perforé ligne à ligne sur ruban.
- Monter et charger la 2^e partie du compilateur (environ 100 m). Déposer le ruban.
- Monter le ruban perforé portant le code intermédiaire. Relancer le programme.
- Le code intermédiaire est lu bloc par bloc, le code binaire est perforé au fur et à mesure.

Exécution

- Lire le ruban perforé « chargeur » de code binaire.
- Monter le ruban perforé portant le code binaire. Lancer le chargeur.
- Monter le ruban perforé portant les routines temps réel (environ 50 m). Relancer le chargeur qui lit les blocs utiles.
- Lancer l'exécution !

Ce n'était pas vraiment complexe, un peu lassant bien sûr, mais très efficace. (De toutes façons nous passions nos

jours à enrouler des rubans perforés !

Mes premières applications

Je n'arrive plus à me souvenir de tout premier programme, mais je sais que comme pour chaque nouveau langage, j'ai immédiatement commencé par un cas concret, directement utile.

Par contre je me rappelle très bien du programme destiné à mesurer le temps disponible lors de l'interruption « coupure secteur » du CAE 133 : un programme assembleur sur la machine en test perforait au fil des essais un ruban perforé avec les valeurs mesurés. Un programme Fortran relisait ensuite le ruban et calculait les valeurs max et min et la moyenne.

L'année suivante, j'ai eu à mener une étude sur des méthodes câblées de calcul des fonctions trigonométriques pour le développement d'une machine du plan calcul. J'avais trouvé une étude intéressante : The CORDIC Trigonometric Computing Technique ⁽²⁾. Quelques bouts de programme Fortran sur CAE 133 m'ont permis de faire rapidement quelques essais.

Vous avez ci joint un extrait du listing source d'époque. Notez l'usage curieux du « \$ » comme séparateur de chaînes de caractères, et la barre de fraction dans les dimensions des tableaux.

Les difficultés ?

Ma foi, c'était facile... sauf quand la machine à écrire se mettait à sauter des

```

CORDIC - COEFFICIENT K (A)
C K= sqrt(1+2**N-2), sqrt(1+2**N-2)...sqrt(1+2**N-2(I-2))
REAL K, IK
DIMENSION KK(1/33), KIK(1/33)
TYPE 1
1 FORMAT (//15X, $CORDIC$, 5X, $COEFFICIENT K$/)
A=1
K=1
DO 10 I=2, 32
B=SQRT (1+A)
K=K*B
A=A/4
CALL FRABINA (K, KK)
IF (SENSE SWITCH 1) 5, 2
2 IF (SENSE SWITCH 2) 7, 3
3 IF (SENSE SWITCH 3) 8, 4
4 PAUSE 77
GO TO 10
5 TYPE 6, I, K, KK
6 FORMAT (12, 2X, E15.9, 4X, I1, $. $, 8(4I1, 1X))
GO TO 10
7 IK=1/K
CALL FRABINA (IK, KIK)
TYPE 6, I, IK, KIK
GO TO 10
8 TYPE 9, I, A, B, K
9 FORMAT (12, 3(4X, E15.9))
10 CONTINUE
** END

```

les valeurs A imprimées sur une seule ligne

lignes, ou des pages... Il m'a fallu interviewer le responsable programmation qui se souvenait d'avoir entendu dire que le premier caractère de chaque ligne commandait la mise en page pour commencer à comprendre. Ce « petit détail » était complètement omis dans la notice dont je disposais !

Et ensuite ?

Et bien à vrai dire, ce programme « cordic » est peut-être le programme le plus « mathématique » que j'ai développé en Fortran.

Pendant plusieurs années chez Philips Data Systems, j'ai aligné des milliers de ligne Fortran pour des programmes de conception assistée par ordinateur : listes de câblage, simulation logique... Très peu de calculs, mais énormément de comparaison de chaînes de caractères (noms de signaux), nous étions spécialistes des « bouclettes » !

C'est à cette époque que j'ai trouvé ma première erreur de code dans une compilation. Ça m'a obligé à analyser le code

généralisé par le compilateur et à découvrir qu'en somme c'était tout simple et très efficace.

J'ai travaillé ensuite dans les filiales télécommunications du groupe, il n'était pas quand même pas question de programmer nos équipements temps réels en Fortran. Mais j'avais toujours le Fortran sous la main pour réaliser rapidement les outils dont nous avons besoin. Par exemple :

- Dés-assembleur pour CP50 (les premiers commutateurs du réseau Transpac) pour décoder des programmes de test dont nous avons perdu les sources.
- Édition de la documentation à partir des commentaires inclus dans le source d'un programme.
- Mise en forme du listing issu d'un macro-assembleur.

Mais le plus beau travail réalisé dans ce genre est sans doute l'analyseur syntaxique PILOTE développé en 1982 à partir d'un article que j'avais repéré en 1969 et soigneusement conservé⁽³⁾.

D'accord, à l'époque les outils LEX et YACC existaient déjà dans le monde UNIX, mais le monde industriel était très loin du monde de la recherche, et ce développement nous a bien servi.

La première application a été l'écriture autour de PILOTE du méta-compilateur MÉTACOMP qui permettait de créer les tables d'analyse de PILOTE à partir d'une description des règles en langage source. Application en circuit fermé en somme ! Mais qui nous a permis ensuite de développer facilement un compilateur du langage CCITT « LDS » (Langage de Description Système) qui nous fournis-

sait automatiquement les automates en code assembleur Z80 pour nos protocoles de télécommunications.

Ont suivi un automate d'analyse de RHM (Relation Homme Machine) pour l'exploitation d'un système de commutation et un langage de test.

Pour les curieux, l'ensemble des sources PILOTE et MÉTACOMP est toujours entre mes mains...

En conclusion

Honnêtement je ne peux pas dire que je considère Fortran comme un « beau » langage, mais ça a toujours été pour moi un outil sûr et extrêmement pratique. Avec les compilateurs dont je disposais, il était facile de comprendre quel code allait être généré, et donc d'optimiser suffisamment sa programmation pour obtenir un programme efficace. Pour moi, ce n'était peut-être que le « couteau suisse » de la programmation, mais c'était le couteau qu'on a toujours en poche et qui marche à tout coup.

Et c'est le premier langage évolué que j'ai découvert. J'y suis resté fidèle jusqu'à la découverte d'APL, j'ai alors développé une passion immédiate pour ce « langage qui rend intelligent » et j'ai délaissé Fortran...

Ph. Denoyelle

Notes :

- (1) voir « Du CAE130 à IRIS65M - 6 ans de la division militaire de CAE-CII » dans les Actes du 6e colloque de l'Histoire de l'Informatique)
- (2) (J.E. Volder) dans les «IRE Transactions on electronic computers»
- (3) « Principes d'une méthode syntaxique d'écriture de compilateurs » (B. Kahan et H. Dumas-Primbault) Revue R.I.R.O. B-2 1969

Downward to (la descente aux enfers vers) Fortran

Tout le monde n'a pas apprécié FORTRAN... Ici J. André nous fait part de son expérience de chercheur.

C'est en 1962-1963, dans le cadre du troisième cycle d'analyse numérique de Jean Legras à Nancy, que j'ai découvert l'« informatique » qu'on appelait encore « le calcul numérique ». Nous disposions alors d'un IBM 650 que l'on programait en assembleur qui faisait l'objet de cours et TP. Cette même année là le programme des cours fut brutalement modifié : en rentrant d'un séminaire organisé à Grenoble par Bolliet et son équipe, Legras et Pair décidèrent de nous enseigner Algol 60, mais bien sûr sans possibilité de compiler nos programmes.

Durant l'été, j'ai eu l'honneur d'être accepté au CERN pour y faire un stage dans la Data Division, celle qui s'occupait des calculs. L'ordinateur du CERN était alors un IBM 709 qui évidemment me paraissait un monstre par rapport au petit 650 de notre université. Ma première réaction a bien sûr été de vouloir y passer des programmes Algol. « Algol, c'est quoi ça ? Ici tout le monde programme en Fortran ou alors en assembleur ». Force m'a donc été d'apprendre Fortran, ce qui était quand même plus simple qu'Algol. Mes premiers essais ont été catastrophiques. J'avais évidemment voulu traduire des programmes d'Algol en Fortran et je me suis fait traité de *frenchie* (car tout le monde ou presque était anglais) : mes premiers programmes calculaient $n!$ avec une procédure récursive et je ne sais plus quoi avec des tableaux

à bornes dynamiques qu'on manipulait avec une boucle dont la limite était n . Pourtant rien ne disait dans le manuel que c'était interdit. J'ai alors tellement ramené Fortran à un niveau zéro de la programmation que j'ai failli me faire virer quand j'ai demandé, puisqu'on ne pouvait rien faire de naturel, comment convertir un réel en entier (si vous avez oublié, était entière toute variable dont l'identificateur commençait par I...N et réelles les autres) et il n'était non plus dit nulle part qu'on pouvait faire $I=N!$

Je ne me suis jamais remis de ce « nivellement par la base », ce retour à un petit nègre alors que je venais d'apprendre une belle langue. Je me suis mis alors à l'assembleur que j'ai d'ailleurs utilisé longtemps pour écrire des compilateurs de langages structurés, considérant avec quelque mépris que si on pouvait tout faire avec Fortran, il fallait tout faire, et qu'on pouvait même faire des erreurs catastrophiques. J'ai eu ainsi quelques années plus tard une sorte de revanche : j'avais participé à l'écriture d'un compilateur Algol optimisé (traitant de façon statique ce qui pouvait l'être) et le même CERN (ou plutôt quelqu'un qui travaillait au CERN) nous envoya un programme Fortran réécrit en Algol comme benchmark de notre compilateur. Nous fûmes surpris de voir que notre optimiseur sortait hors des boucles pratiquement tous les calculs qui devenaient alors très statiques et rapides. De fait beaucoup de variables étaient définies à la racine du programme et en regardant de plus près le code Fortran, nous avons vu que

les auteurs du programme avaient voulu simuler une sorte de passage de paramètres par nom et s'étaient plantés en utilisant les mêmes noms de variables (du type I ou N) dans des routines qui s'appelaient mutuellement, ce qui donnait, même en Fortran, des résultats complètement faux. Le concept de portée ne faisait pas non plus partie de Fortran. On n'a jamais su si ce programme était important dans le contrôle de la radioactivité au CERN.

Quarante ans plus tard, je ne fais plus de compilateurs mais m'occupe de documents numériques. E je retrouve dans Word tout ce que j'ai toujours détesté dans Fortran et prend mon pied avec LaTeX ou XML comme je l'ai pris avec Algol!

Jacques.Andre@irisa.fr

En direct du labo :

*La théorie, c'est quand on sait tout
et que rien ne fonctionne.*

*La pratique, c'est quand tout fonctionne
et que personne ne sait pourquoi.*

*Ici, nous avons réuni théorie et pratique :
rien ne fonctionne...
et personne ne sait pourquoi !*

Une curiosité : la variation des constantes

En Fortran, le passage des paramètres aux sousroutines se fait « par adresse ». Cela peut provoquer de jolies surprises :

Écrivons :

```
SUBROUTINE PLUS (A)
WRITE 1,100 (A)
100  FORMAT (1X, I6)
A=A+1
RETURN
```

Dans le programme principal appelons notre sousroutine plusieurs fois de suite

```
CALL PLUS (123)
CALL PLUS (123) ... ..
```

Et qu'obtenons nous ? Et bien très naturellement 123, 124, 125, 126, 127...

En effet, le compilateur range la constante en mémoire et fournit l'adresse de cette case mémoire à la sousroutine. Celle-ci se fait alors un devoir d'incrémenter la constante...

Ceci était encore vrai avec le Fortran IV de Philips, mais plus avec le Fortran 77 de DEC...

Ph. Denoyelle

Recevoir une copie électronique du bulletin ?

Si vous le souhaitez, nous pouvons vous envoyer le « Bulletin de l'Aconit » sous forme électronique (fichier PDF).

Si cette diffusion « express » vous intéresse, merci de le signaler à Flore <flore.gully@aconit.org>, et l'association fera quelques économies de timbres...