



HAL
open science

Langage et système d'exploitation pour expériences nucléaires

Y. Beauval, J.P. Delbreil, J.P. Chardenon, J.P. Meinadier

► **To cite this version:**

Y. Beauval, J.P. Delbreil, J.P. Chardenon, J.P. Meinadier. Langage et système d'exploitation pour expériences nucléaires. *Revue de Physique Appliquée*, 1969, 4 (2), pp.133-134. 10.1051/rphysap:0196900402013300 . jpa-00243179

HAL Id: jpa-00243179

<https://hal.science/jpa-00243179>

Submitted on 4 Feb 2008

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

LANGAGE ET SYSTÈME D'EXPLOITATION POUR EXPÉRIENCES NUCLÉAIRES

Y. BEAUVAL, J. P. DELBREIL, J. P. CHARDENON et J. P. MEINADIER,

Département de Calcul Électronique, C.E.N., Saclay.

Résumé. — On décrit un langage interprétatif orienté vers la physique nucléaire et le système d'exploitation associé qui permet son utilisation en mode programmé et en mode conversationnel.

Abstract. — A software system for small computers is described. It allows an interpretative language for pulse height analysis that can be used in a conversational way as well as in a programmed way.

Notre but était de définir et d'étudier un système d'exploitation applicable à un petit ensemble de traitement de l'information, destiné à être industrialisé pour des applications d'expériences nucléaires et de spectrométrie. Notre recherche est partie de deux idées : *i*) permettre au physicien de ne connaître qu'un seul langage pour, d'une part, programmer l'acquisition, le traitement, la visualisation de ses expériences et, d'autre part, les dépouiller en mode conversationnel; *ii*) limiter au maximum les contingences de la programmation : le physicien ne se préoccupe ni de la segmentation de ses programmes, ni de l'emplacement en mémoire rapide ou auxiliaire des spectres sur lesquels il travaille. Nous allons décrire un premier système (langage et système d'exploitation) réalisé à titre expérimental sur un calculateur C 90-40.

I. Le langage. — Spécialement adapté à la manipulation des spectres, il n'est pas figé et peut facilement être complété par de nouvelles instructions. Il opère sur des scalaires (variables ou constantes) et des spectres, soit au niveau du spectre entier : A ; soit au niveau d'une portion de spectre : $A((I, J))$, soit au niveau d'une composante : $A(K)$. Il possède les 4 opérations. Si celles-ci portent sur deux spectres, les opérations sont réalisées canal par canal. Par exemple : $A = K * A$: réalise une affinité de facteur K ; $A = B - C$: réalise une soustraction canal à canal.

Des fonctions permettent de résoudre des problèmes courants en physique nucléaire. Par exemple :

$$B = AIRE(A((I, J)))$$

réalise la somme des canaux du spectre : A : entre les canaux I et J ,

$$B = LISS(N, A)$$

réalise le lissage du spectre : A : par moyenne sur N canaux.

Le langage possède de plus les instructions de branchements conditionnels ou inconditionnels nécessaires, les instructions d'entrée-sortie et les pseudo-instructions à l'usage des différents éléments du système d'exploitation.

II. Le système d'exploitation. — Il se caractérise principalement par deux aspects : 1) l'interprétation et l'exécution des instructions directement à partir du langage source. Chaque instruction de ce langage s'exécute sous forme d'une séquence d'instruction en langage machine analogue à une macro-instruction. Ceci permet l'utilisation du langage aussi bien en mode programmé qu'en mode pas à pas; 2) l'utilisation systématique de la mémoire auxiliaire qui contient, au moins dans les petits systèmes, le programme source, la plupart des macro-instructions, les spectres.

Le système se compose des éléments suivants : D'une part, un éditeur éventuellement non résident qui permet l'édition et le rangement des programmes en mémoire auxiliaire en se chargeant de la gestion des étiquettes et de l'édition des diagnostics de syntaxe. D'autre part, un ensemble partiellement résident comprenant : *i*) un superviseur qui établit le lien entre les événements extérieurs (demande de mode conversationnel, fin d'opération d'entrée-sortie) et les différentes tâches en mémoire, et comprend notamment la gestion de la machine à écrire et la gestion de la mémoire auxiliaire; *ii*) un séquenceur de gestion du programme source réalisant automatiquement la segmentation du programme (dont seules quelques instructions sont en mémoire centrale) prenant notamment en charge les ruptures de séquences et la gestion des boucles; *iii*) un interpréteur d'instruction de langage source, qui analyse l'instruction du langage source, complète éventuellement les diagnostics fournis par l'éditeur, charge en un emplacement fixe (afin d'éviter l'entrée

en translatable) la macro-instruction correspondante si elle n'est pas résidente et, avant de lui donner le contrôle, passe la main aux deux programmes chargés de consigner en emplacement fixe les adresses des opérands utilisées et qui sont :

- le programme de gestion des scalaires qui gère la table des symboles scalaires et génère les scalaires constants;
- le programme de gestion des spectres qui gère la table des adresses d'implantation des spectres dans les mémoires ainsi que les zones de mémoire centrale et auxiliaires qui leur sont réservées. Dans le cas où la place en mémoire centrale est insuffisante, ce programme assure également la segmentation des spectres.

Les idées contenues dans cette communication ont été à la base de notre collaboration avec la Société Intertechnique pour la conception du langage « Phitrol » et du système d'exploitation « Sel 1 » de l'ensemble de contrôle d'expériences nucléaires « Nudiac ». « Phitrol » comporte de plus des instructions de démarrage d'acquisition en temps réel qui contiennent des indications de chaînage permettant, en fin d'acquisition, l'appel automatique du programme de dépouillement associé.

En conclusion, l'emploi d'une technique d'interprétation plutôt que d'une technique de compilation donne à un petit système des possibilités de langage et de facilité d'utilisation analogues à celles présentées par un gros calculateur, au détriment, évidemment, des performances au niveau de l'exécution des programmes.

MÉTHODE D'ACCÈS RAPIDE AUX DÉPASSEMENTS DE CAPACITÉ DANS LES PETITS CALCULATEURS UTILISÉS COMME ANALYSEURS MULTICANAUX : L'ADRESSAGE INDIRECT PROGRAMME

R. CHAMINADE, J. PAIN, M. LECHACZYNSKI, J. C. FAIVRE,
J. DEMEUZOIS et J. DURRUTY,

Service de Physique Nucléaire à Moyenne Énergie, C.E.N., Saclay.

Résumé. — Pour un calculateur ayant une mémoire de 12 bits par exemple, la capacité d'une adresse est limitée à 4095 événements. Pour accroître la capacité des canaux en dépassement, on utilise généralement une table associative des dépassements. Mais la recherche d'un canal particulier, dans cette table, pour visualisation ou mise à jour, est un processus lent. Pour y remédier, nous utilisons l'adressage indirect programmé pour l'accès à un canal particulier dans la table double précision remplaçant la table associative. Nous limitons la capacité des canaux à 2047, soit 11 bits, pour réserver le douzième bit, bit supérieur, comme bit d'adressage indirect. Quand le douzième bit est à 0, le comptage du canal est représenté par les 11 premiers bits. Si le comptage dépasse 2047 événements, le douzième bit est imposé en permanence à 1, les 11 premiers bits représentent alors l'adresse (ou l'index d'accès) dans la table en double précision où se trouve maintenant le comptage. Le calculateur PDP 8 comporte une instruction de test du douzième bit.

Abstract. — In a small digital computer having, for example, 12 bits memory, the storage capacity is limited to 4095 events. To increase the capacity for channels with overpasses, one ordinarily uses an associative overflow table, but the search in this table of a particular channel, for display or up-dating purposes, is slow. To cure that we use software indirect addressing to access a particular channel in the double precision overflow table which replaces the associative overflow table. We limit the channel capacity to 2047, or 11 bits, to keep the 12th upper bit as an indirect address bit. If the 12th bit is 0, the channel content is the channel count as normally. But if the count overpasses 2047 the 12th bit is set and maintained to 1, the count is then replaced, in the channel, by the address (or the index) in the double precision table where one now stores this count. The PDP 8 computer has an instruction testing this 12th bit.