

INTERNATIONAL STANDARD NORME INTERNATIONALE



2382 / VII

INTERNATIONAL ORGANIZATION FOR STANDARDIZATION • МЕЖДУНАРОДНАЯ ОРГАНИЗАЦИЯ ПО СТАНДАРТИЗАЦИИ • ORGANISATION INTERNATIONALE DE NORMALISATION

Data processing — Vocabulary — Section 07 : Digital computer programming

First edition — 1977-03-01

Traitement de l'information — Vocabulaire — Chapitre 07 : Programmation des calculateurs numériques

Première édition — 1977-03-01

UDC/CDU 681.3 : 001.4

Ref. No./Réf. n° : ISO 2382/VII-1977 (E/F)

Descriptors : data processing, programming (computers), vocabulary/**Descripteurs** : traitement de l'information, programmation des calculateurs, vocabulaire.

Price based on 36 pages/Prix basé sur 36 pages

FOREWORD

ISO (the International Organization for Standardization) is a worldwide federation of national standards institutes (ISO member bodies). The work of developing International Standards is carried out through ISO technical committees. Every member body interested in a subject for which a technical committee has been set up has the right to be represented on that committee. International organizations, governmental and non-governmental, in liaison with ISO, also take part in the work.

Draft International Standards adopted by the technical committees are circulated to the member bodies for approval before their acceptance as International Standards by the ISO Council.

International Standard ISO 2382/VII was developed by Technical Committee ISO/TC 97, *Computers and information processing*, and was circulated to the member bodies in August 1974.

It has been approved by the member bodies of the following countries :

Australia	Italy	Turkey
Belgium	New Zealand	United Kingdom
Brazil	Poland	U.S.A.
Czechoslovakia	Romania	U.S.S.R.
France	South Africa, Rep. of	Yugoslavia
Hungary	Sweden	
Ireland	Switzerland	

The member body of the following country expressed disapproval of the document on technical grounds :

Germany

AVANT-PROPOS

L'ISO (Organisation internationale de normalisation) est une fédération mondiale d'organismes nationaux de normalisation (comités membres de l'ISO). L'élaboration des Normes internationales est confiée aux comités techniques de l'ISO. Chaque comité membre intéressé par une étude a le droit de faire partie du comité technique correspondant. Les organisations internationales, gouvernementales et non gouvernementales, en liaison avec l'ISO, participent également aux travaux.

Les projets de Normes internationales adoptés par les comités techniques sont soumis aux comités membres pour approbation, avant leur acceptation comme Normes internationales par le Conseil de l'ISO.

La Norme internationale ISO 2382/VII a été élaborée par le comité technique ISO/TC 97, *Calculateurs et traitement de l'information*, et a été soumise aux comités membres en août 1974.

Les comités membres des pays suivants l'ont approuvée :

Afrique du Sud, Rép. d'	Italie	Tchécoslovaquie
Australie	Nouvelle-Zélande	Turquie
Belgique	Pologne	U.S.A.
Brésil	Roumanie	U.R.S.S.
France	Royaume-Uni	Yougoslavie
Hongrie	Suède	
Irlande	Suisse	

Le comité membre du pays suivant l'a désapprouvée pour des raisons techniques :

Allemagne

Data processing gives rise to numerous international exchanges of both intellectual and material nature. These exchanges often become difficult, either because of the great variety of terms used in various fields or languages to express the same concept, or because of the absence of or the imprecision of useful concepts.

To avoid misunderstandings due to this situation and to facilitate such exchanges, it is advisable to select terms to be used in various languages or in various countries to express the same concept and to establish definitions providing satisfactory equivalents for the various terms in different languages.

In accordance with the directions given to the ISO Sub-Committee in charge of the Vocabulary, the work on it has been mainly based on the usage to be found in the *Vocabulary of information processing** established and published by the International Federation for Information Processing and the International Computation Centre, and in the *USA Standard vocabulary for information processing* established, published, and revised by the American National Standards Institute. The Sub-Committee also considered various international documents or drafts issued by ISO Technical Committee 97 and its Sub-Committees and other international organizations (such as the International Telecommunication Union) and national drafts or standards.

The definitions have been drawn up with the objective of achieving a proper balance between precision and simplicity. The main objective of this Vocabulary is to provide definitions that can be understood to have the same meaning by all concerned. It may thus be felt that some definitions are not sufficiently precise, do not include all cases, do not take into account certain exceptions, or are in conflict with established uses in particular fields of application.

In addition, the Vocabulary consists of several sections prepared over a long period of time and it may be that the preparation of the later sections introduces inconsistencies with the earlier ones.

These imperfections will be eliminated as far as possible in later editions. This procedure allows for immediate publication of needed sections and permits an element of flexibility in the preparation of a comprehensive vocabulary in view of the dynamics of language.

* North Holland Publishing Company — AMSTERDAM 1966.

Le traitement de l'information donne lieu à de très nombreux échanges internationaux d'ordre intellectuel ou matériel qui sont souvent rendus difficiles soit par la diversité des termes employés dans différents milieux ou dans différentes langues pour exprimer une même notion, soit par l'absence ou l'imprécision des définitions des notions utiles.

Pour éviter les malentendus ayant leur origine dans le vocabulaire et faciliter les échanges, il convient de procéder à un choix des termes à employer dans les différentes langues ou dans les différents pays pour désigner la même notion, et de rédiger des définitions assurant une équivalence pratiquement satisfaisante entre ces différents termes.

Conformément aux directives reçues par le sous-comité de l'ISO chargé de l'étude du Vocabulaire, les travaux correspondants ont été essentiellement basés sur l'usage codifié dans le *Vocabulary of information processing** établi et publié par l'International Federation for Information Processing et le Centre international de calcul, et dans le *USA Standard vocabulary for information processing* établi, publié et révisé par l'American National Standards Institute. Le sous-comité s'est appuyé en outre sur différents documents ou projets internationaux issus du Comité technique 97 de l'ISO et de ses sous-comités ou d'autres organisations internationales (telles que l'Union internationale des télécommunications), ainsi que sur des normes ou projets nationaux.

Les définitions ont été conçues de façon telle qu'un équilibre raisonnable entre la précision et la simplicité soit atteint. L'objectif principal de ce Vocabulaire est de fournir des définitions qui puissent être reconnues comme ayant le même sens par tout lecteur concerné. Quelques définitions peuvent donc sembler insuffisamment précises, ne pas inclure tous les cas, ne pas tenir compte de certaines exceptions ou être en contradiction avec les usages établis dans des domaines d'application particuliers.

De plus, le Vocabulaire est constitué de plusieurs chapitres dont l'élaboration s'est étalée sur une grande période de temps et la réalisation de nouveaux chapitres peut introduire des incohérences dans les anciens chapitres.

Ces imperfections seront éliminées dans la mesure du possible dans les éditions ultérieures. Cette procédure permet de publier rapidement les chapitres les plus attendus et introduit un élément de souplesse dans la réalisation d'un vocabulaire étendu et devant s'adapter à la dynamique de la langue.

* North Holland Publishing Company — AMSTERDAM 1966.

CONTENTS

	Page
1 General	1
1.1 Introduction	1
1.2 Scope	1
1.3 Field of application	1
2 Principles and rules followed	1
2.1 Definition of an entry	1
2.2 Organization of an entry	1
2.3 Classification of entries	1
2.4 Selection of terms and wording of definitions	1
2.5 Multiple meanings	1
2.6 Abbreviations	1
2.7 Use of parentheses	2
2.8 Use of (square) brackets	2
2.9 Use of terms printed in italic typeface in definitions and use of asterisks	2
2.10 Spelling	2
2.11 Organization of the alphabetical index	2
3 Terms and definitions	
07 Digital computer programming	2
07.01 General terms	2
07.02 Languages	4
07.03 Translation methods	6
07.04 Data description in a language	8
07.05 Program variables and parameters	9
07.06 Organizational programs and techniques	10
07.07 Utility routines and techniques	11
07.08 Subroutines and multiple use programs	12
07.09 Entry points, exits and links between programs	13
07.10 Loops	15
07.11 Jumps and branching	16
07.12 Storage allocations and segmentation of programs	16
07.13 Interruptions and restarts	17
07.14 Dumping	18
07.15 Program maintenance	19
07.16 Instructions — General terms	20
07.17 Instruction formats	21
07.18 Types of instructions	22
07.19 Addresses	23
07.20 Addressing methods	25
4 Alphabetical index	
English	26
French	32

SOMMAIRE

	Page
1 Généralités	1
1.1 Introduction	1
1.2 Objet	1
1.3 Domaine d'application	1
2 Principes d'établissement et règles suivies	1
2.1 Définition de l'article	1
2.2 Constitution d'un article	1
2.3 Classification des articles	1
2.4 Choix des termes et des définitions	1
2.5 Pluralité de sens ou polysémie	1
2.6 Abréviations	1
2.7 Emploi des parenthèses	2
2.8 Emploi des crochets	2
2.9 Emploi dans les définitions de termes écrits en caractères italiques et de l'astérisque	2
2.10 Mode d'écriture et orthographe	2
2.11 Constitution de l'index alphabétique	2
3 Termes et définitions	
07 Programmation des calculateurs numériques	3
07.02 Langages	4
07.03 Méthodes de traduction	6
07.04 Description des données dans un langage	8
07.05 Variables et paramètres du programme	9
07.06 Techniques et programmes d'organisation	10
07.07 Techniques et programmes de service	11
07.08 Programmes à usage multiple et sous-programmes	12
07.09 Débuts et fins de programmes et liens entre programmes	13
07.10 Boucles	15
07.11 Sauts et aiguillages	16
07.12 Attribution de mémoire et segmentation des programmes	16
07.13 Interruptions et reprises	17
07.14 Vidages	18
07.15 Maintenance des programmes	19
07.16 Instructions — Termes généraux	20
07.17 Clichés d'instruction	21
07.18 Types d'instructions	22
07.19 Adresses	23
07.20 Méthodes d'adressage	25
4 Index alphabétique	
Anglais	26
Français	32

STANDARDSISO.COM : Click to view the full PDF of ISO 2382-7:1977

**Data processing — Vocabulary —
Section 07 : Digital computer
programming**

1 GENERAL

1.1 Introduction

This section of the Vocabulary (which will comprise some twenty sections) deals with concepts on digital computer programming.

1.2 Scope

The Vocabulary is intended to facilitate international communication in data processing. It presents, in two languages, terms and definitions of selected concepts relevant to the field of data processing and identifies relationships between the entries.

In order to facilitate their translation into other languages, the definitions are drafted so as to avoid, as far as possible, any peculiarity attached to a language.

1.3 Field of application

The Vocabulary deals with the main areas of data processing, including the principal processes and types of equipment used, the representation, organization and presentation of data, the programming and operation of computers, input-output devices and peripheral equipment, as well as particular applications.

2 PRINCIPLES AND RULES FOLLOWED

The sub-clauses under this heading, included in ISO 2382/I, are equally applicable to this section. They are not reproduced here. The corresponding sub-clause headings are the following :

2.1 Definition of an entry

2.2 Organization of an entry

2.3 Classification of entries

2.4 Selection of terms and wording of definitions

2.5 Multiple meanings

2.6 Abbreviations

**Traitement de l'information —
Vocabulaire —
Chapitre 07 : Programmation des
calculateurs numériques**

1 GÉNÉRALITÉS

1.1 Introduction

Le présent chapitre du Vocabulaire (qui comprendra une vingtaine de chapitres) contient les notions les plus courantes intéressant la programmation des calculateurs numériques.

1.2 Objet

Le Vocabulaire a pour objet de faciliter les échanges internationaux dans ce domaine. Il présente un ensemble bilingue de termes et de définitions ayant trait à des notions choisies, et définit les relations pouvant exister entre différentes notions.

Les définitions ont été établies de manière à ne présenter que peu de particularités attachées à une langue donnée, en vue de faciliter leur transposition dans d'autres langues.

1.3 Domaine d'application

Le Vocabulaire traite des principaux domaines du traitement de l'information, des principaux procédés et types de machines employés, de la représentation et de la forme des données, de la programmation et de l'exploitation des calculateurs, des entrées-sorties et organes périphériques, et de certaines applications.

2 PRINCIPES D'ÉTABLISSEMENT ET RÈGLES SUIVIES

Les textes des paragraphes ci-dessous, inclus dans l'ISO 2382/I, s'appliquent également au présent chapitre. Ils ne sont pas reproduits ici. Les titres des paragraphes correspondants sont les suivants :

2.1 Définition de l'article

2.2 Constitution d'un article

2.3 Classification des articles

2.4 Choix des termes et des définitions

2.5 Pluralité de sens ou polysémie

2.6 Abréviations

2.7 Use of parentheses	2.7 Emploi des parenthèses
2.8 Use of (square) brackets	2.8 Emploi des crochets
2.9 Use of terms printed in italic typeface in definitions and use of asterisks	2.9 Emploi dans les définitions de termes écrits en caractères italiques et de l'astérisque
2.10 Spelling	2.10 Mode d'écriture et orthographe
2.11 Organization of the alphabetical index	2.11 Constitution de l'index alphabétique

STANDARDSISO.COM : Click to view the full PDF of ISO 2382-7:1977

3 TERMS AND DEFINITIONS

07 DIGITAL COMPUTER PROGRAMMING

07.01 GENERAL TERMS

07.01.01

programming system

One or more *programming languages* and the necessary *software* for using these languages with particular *automatic data processing* equipment.

07.01.02

(program) library

An organized collection of *computer programs*.

07.01.03

library program

A *computer program* in or from a *program library*.

07.01.04

(language) processor

A *computer program* that performs such functions as *translating*, * *interpreting* and other tasks required for processing a specified *programming language*.

Exemples : A FORTRAN processor; a COBOL processor.

07.01.05

preprocessor

A *computer program* that effects some preliminary computation or organization.

07.01.06

postprocessor

A *computer program* that effects some final computation or organization.

07.01.07

statement

In a *programming language*, a meaningful expression that may describe or specify *operations* and is usually complete in the context of this language.

07.01.08

comment**(computer program) annotation****remark****note**

A description, reference or explanation, added to or interspersed among the *statements* of the *source language*, that has no effect in the *target language*.

3 TERMES ET DÉFINITIONS

07 PROGRAMMATION DES CALCULATEURS NUMÉRIQUES

07.01 TERMES GÉNÉRAUX

07.01.01

système de programmation

Ensemble comprenant un ou plusieurs *langages de programmation* ainsi que le *logiciel* nécessaire pour l'emploi de ces langages sur un matériel particulier de *traitement automatique de l'information*.

07.01.02

bibliothèque (de programmes) programmathèque

Ensemble organisé de *programmes de calculateur*.

07.01.03

programme de bibliothèque

Programme de calculateur appartenant à une *bibliothèque de programmes*.

07.01.04

processeur (de langage)

Programme de calculateur qui assure des fonctions telles que *traduire*, * *interpréter*, et d'autres tâches nécessaires pour le traitement d'un *langage de programmation* donné.

Exemples : Un processeur FORTRAN; un processeur COBOL.

07.01.05

préprocesseur

Programme de calculateur qui effectue certaines tâches préliminaires de calcul ou d'organisation.

07.01.06

postprocesseur

Programme de calculateur qui effectue certaines tâches de calcul ou d'organisation en fin de travail.

07.01.07

1 instruction

Dans un *langage de programmation*, expression possédant une signification, qui peut décrire ou spécifier des *opérations* et qui forme généralement un tout dans ce langage.

07.01.08

commentaire

Note descriptive, explicative, ou de référence, ajoutée ou intercalée parmi les *instructions* en *langage d'origine*, et sans effet en *langage résultant*.

07.01.09

instruction

statement (deprecated in this sense)

order (deprecated in this sense)

command (deprecated)

In a *programming language*, a meaningful expression that specifies one *operation* and identifies its *operands*, if any.

07.01.10

declaration

directive

declarative (deprecated)

pseudo-instruction (obsolescent and deprecated)

In a *programming language*, a meaningful expression that affects the interpretation of other expressions in that language.

07.01.11

address

A *character* or group of characters that identifies a *register*, a particular part of *storage*, or some other *data* source or destination.

07.01.12

minimum delay programming

A method of *programming* in which *storage locations* for *instructions* and *data* are chosen so that *access time* is reduced and minimized.

07.01.13

microprogram

A *sequence* of elementary *instructions* that corresponds to a specific *computer operation*, that is maintained in special *storage*, and whose *execution* is initiated by the introduction of a *computer instruction* into an *instruction register* of a *computer*.

07.01.14

microprogramming

The *preparation* or use of *microprograms*.

07.02 LANGUAGES

07.02.01

language

A *set* of *characters*, conventions, and rules that is used for conveying *information*.

07.02.02

natural language

A *language* whose rules are based on current usage without being explicitly prescribed.

07.01.09

2 instruction

Dans un *langage de programmation*, expression possédant une signification, qui spécifie une *opération* et, le cas échéant, identifie les *opérandes*.

07.01.10

pseudo-instruction

directive

Dans un *langage de programmation*, expression possédant une signification, qui affecte l'interprétation d'autres expressions écrites dans ce langage.

07.01.11

adresse

Caractère ou groupe de caractères qui identifie un *registre*, une partie déterminée d'une *mémoire*, ou quelque autre origine ou destination de *données*.

07.01.12

programmation à temps d'exécution minimal

Méthode de *programmation* dans laquelle les *emplacements de mémoire* des *instructions* et des *données* sont choisis de façon à réduire le plus possible les *temps d'accès*.

07.01.13

microprogramme

Suite d'*instructions* élémentaires qui correspond à une *opération de calculateur* particulière, qui est maintenue dans une *mémoire spéciale*, et dont l'*exécution* est déclenchée par l'introduction d'une *instruction-machine* dans le *registre d'instruction* d'un *calculateur*.

07.01.14

microprogrammation

Préparation ou utilisation de *microprogrammes*.

07.02 LANGAGES

07.02.01

langage

Ensemble de *caractères*, de conventions ou de règles employé pour communiquer des *informations*.

07.02.02

langage naturel

Langage dont les règles sont déduites de l'usage courant plutôt que fixées d'une manière formelle.

07.02.03
artificial language
A *language* whose rules are explicitly established prior to its use.

07.02.04
syntax
The relationships among *characters* or groups of characters, independent of their meanings or the manner of their interpretation and use.

07.02.05
semantics
The relationships of *characters* or groups of characters to their meanings, independent of the manner of their interpretation and use.

07.02.06
pragmatics
The relationships of *characters* or groups of characters to their interpretation and use.

07.02.07
metalanguage
A *language* that is used to specify a language or languages.

07.02.08
stratified language
A *language* that cannot be used as its own *metalanguage*.
Example : FORTRAN.

07.02.09
unstratified language
A *language* that can be used as its own *metalanguage*.
Example : Most *natural languages*.

07.02.10
source language
A *language* from which *statements* are translated.

07.02.11
target language
object language
A *language* into which *statements* are translated.

07.02.12
algorithmic language
An *artificial language* established for expressing *algorithms*.

07.02.13
programming language
An *artificial language* established for expressing *computer programs*.

07.02.03
langage artificiel
Langage dont les règles ont été fixées explicitement avant toute consécration par l'usage.

07.02.04
syntaxe
Ensemble des relations entre les *caractères* ou groupes de caractères, indépendamment de leurs significations ou de la façon de les employer ou de les interpréter.

07.02.05
sémantique
Ensemble des relations entre les *caractères* ou groupes de caractères, et leur significations, indépendamment de la façon de les employer ou de les interpréter.

07.02.06
pragmatique
Ensemble des relations entre les *caractères* ou groupes de caractères et la signification qui leur est attribuée dans le contexte où ils sont employés.

07.02.07
métalangage
Langage servant à décrire un ou plusieurs langages.

07.02.08
langage stratifié
Langage qui ne peut pas servir de *métalangage* pour lui-même.
Exemple : FORTRAN.

07.02.09
langage non stratifié
Langage qui peut servir de *métalangage* pour lui-même.
Exemple : La plupart des *langages naturels*.

07.02.10
langage d'origine
langage-source
Langage dont on traduit les *instructions*.

07.02.11
langage résultant
langage-objet
Langage dans lequel on traduit les *instructions*.

07.02.12
langage algorithmique
Langage artificiel destiné à exprimer des *algorithmes*.

07.02.13
langage de programmation
Langage artificiel destiné à exprimer des *programmes de calculateur*.

07.02.14
computer-oriented language
low-level language
computer language (deprecated in this sense)
A *programming language* that reflects the structure of a given *computer* or that of a given class of computers.

07.02.15
computer language
machine language
A *computer-oriented language* whose *instructions* consist only of *computer instructions*.

07.02.16
assembly language
A *computer-oriented language* whose *instructions* are usually in one-to-one correspondence with *computer instructions* and that may provide facilities such as the use of *macro-instructions*.

07.02.17
high-level language
A *programming language* that does not reflect the structure of any one given *computer* or that of any one given class of computers.

07.02.18
problem-oriented language
A *programming language* that is especially suitable for a given class of problems.

Exemples :
Procedure-oriented languages such as FORTRAN, ALGOL;
*Simulation * languages* such as GPSS, SIMSCRIPT;
List processing languages such as LISP, IPL-V;
Information retrieval languages.

07.02.19
procedure-oriented language
procedural language
A *problem-oriented language* that facilitates the expression of a *procedure* as an explicit *algorithm*.

Exemples : FORTRAN, ALGOL, COBOL, PL/1.

07.02.20
pseudocode
A *code* that requires *translation* prior to *execution*.

07.03 TRANSLATION METHODS

07.03.01
source program
A *computer program* expressed in a *source language*.

07.03.02
target program
object program
A *computer program* in a *target language* that has been translated from a *source language*.

07.02.14
langage lié au calculateur
Langage de programmation dont la conception est fondée sur la structure d'un *calculateur* donné ou d'une classe donnée de calculateurs.

07.02.15
langage-machine
Langage lié au calculateur, dont toutes les *instructions* sont des *instructions-machine*.

07.02.16
langage d'assemblage
Langage lié au calculateur, dans lequel la plupart des *instructions* sont en correspondance biunivoque avec des *instructions-machine* et qui peut, en outre, comporter d'autres possibilités telles que des *macro-instructions*.

07.02.17
langage évolué
Langage de programmation qui, par conception, n'est lié à la structure d'aucun *calculateur* particulier ou d'aucune classe particulière de calculateurs.

07.02.18
langage adapté aux problèmes
Langage de programmation qui convient particulièrement à un type donné de problèmes.

Exemples :
Langages de procédure tels que FORTRAN, ALGOL;
Langages de simulation tels que GPSS, SIMSCRIPT;
Langages de traitement de listes tels que LISP, IPL-V;
Langages de recherche documentaire.

07.02.19
langage de procédure
langage adapté aux procédures
Langage adapté aux problèmes conçu de façon à faciliter l'expression d'une *procédure* sous forme d'un *algorithme* explicite.

Exemples : FORTRAN, ALGOL, COBOL, PL/1.

07.02.20
pseudocode
Code nécessitant une *traduction* en vue de son *exécution*.

07.03 MÉTHODES DE TRADUCTION

07.03.01
programme d'origine
programme-source
Programme de calculateur écrit en *langage d'origine*.

07.03.02
programme résultant
programme-objet
Programme de calculateur, écrit en *langage résultant*, qui a été traduit d'un *langage d'origine*.

07.03.03
translator
translater
translating program

A *computer program* that *translates* from one language into another language and in particular from one *programming language* into another programming language.

07.03.04
to assemble

To *translate* a *program* expressed in an *assembly language* into a *computer language* and perhaps to link * *subroutines*.

NOTE — Assembling is usually accomplished by substituting the computer language *operation code* for the assembly language operation code and by substituting *absolute addresses*, * *immediate addresses*, * *relocatable addresses*, or *virtual addresses* for *symbolic addresses*.

07.03.05
assembler
assembly program

A *computer program* used to *assemble*.

07.03.06
to compile

To *translate* a *computer program* expressed in a *problem-oriented language* into a *computer-oriented language*.

07.03.07
compiler
compiling program

A *computer program* used to *compile*.

07.03.08
to interpret

To *translate* and to *execute* each *source language* * *statement* of a *computer program* before translating and executing the next statement.

07.03.09
interpreter (in computer programming)
interpretive program

A *computer program* used to *interpret*.

07.03.10
compiler generator

A *translator* or an *interpreter* used to construct *compilers*.

07.03.11
macro generator
macro generating program

A *computer program* that replaces *macroinstructions* in the *source language* with the defined *sequence of instructions* in the source language.

07.03.03
(programme) traducteur
programme de traduction

Programme de calculateur qui *traduit* d'un *langage* vers un autre langage, en particulier d'un *langage de programmation* vers un autre langage de programmation.

07.03.04
assembler

Traduire en *langage-machine* un *programme* écrit en *langage d'assemblage* et éventuellement relier les *sous-programmes*.

NOTE — L'assemblage est effectué notamment en substituant le *code d'opérations* du langage-machine au code d'opérations du langage d'assemblage et en substituant des *adresses absolues*, des *adresses immédiates*, des *adresses translatables* ou des *adresses virtuelles* aux *adresses symboliques*.

07.03.05
assembleur
programme d'assemblage

Programme de calculateur destiné à *assembler*.

07.03.06
compiler

Traduire en *langage lié au calculateur* un *programme de calculateur* exprimé dans un *langage adapté aux problèmes*.

07.03.07
compilateur

Programme de calculateur destiné à *compiler*.

07.03.08
interpréter

Traduire et *exécuter* chaque *instruction* du *langage d'origine* d'un *programme de calculateur* avant de traduire et d'exécuter l'*instruction* suivante.

07.03.09
interpréteur
programme interprétatif

Programme de calculateur destiné à *interpréter*.

07.03.10
générateur de compilateurs

Programme traducteur ou *interpréteur* destiné à construire des *compilateurs*.

07.03.11
macro-générateur

Programme de calculateur destiné à remplacer chaque *macro-instruction* du *langage d'origine* par une suite prédéterminée d'*instructions* du langage d'origine.

07.03.12

translate [compile] [assembly] [execute] phase
translating [compiling] [assembling] [executing] phase
Of a *run*, the logical subdivision that includes the *execution* of the *translator* [the *compiler*] [the *assembler*] [the *target program*].

07.03.13

translating [compiling] [assembling] [running] time
translate [compile] [assemble] [run] duration
The elapsed time taken for the *execution* of a *translator* [a *compiler*] [an *assembler*] [a *target program*].

07.03.12

phase de traduction [de compilation] [d'assemblage] [d'exécution]
Partie d'un *passage* pendant laquelle s'exécute un *programme traducteur* [un *compilateur*] [un *assembleur*] [un *programme résultant*].

07.03.13

durée de traduction [de compilation] [d'assemblage] [d'exécution]
Temps nécessaire à l'exécution d'un *programme traducteur* [d'un *compilateur*] [d'un *assembleur*] [d'un *programme résultant*].

07.04 DATA DESCRIPTION IN A LANGUAGE

07.04 DESCRIPTION DES DONNÉES DANS UN LANGAGE

07.04.01

identifier
A *character* or group of characters used to identify or name an *item* of *data* and possibly to indicate certain properties of that data.

07.04.01

identificateur
Caractère ou groupe de caractères employé pour identifier ou désigner une *donnée* et, éventuellement, mettre en évidence certaines propriétés de cette donnée.

07.04.02

data name
A *character* or group of characters used to identify an *item* of *data*.

07.04.02

nom de donnée
Caractère ou groupe de caractères employé pour identifier une *donnée*.

07.04.03

figurative constant
A *data name* that is reserved for a specified constant in a specified *programming language*.

07.04.03

constante figurative
Nom de donnée réservé à une constante particulière dans un *langage de programmation* particulier.

07.04.04

qualified name
A *data name* explicitly accompanied by a specification of the class to which it belongs in a specified classification system.

07.04.04

nom qualifié
Nom de donnée accompagné de l'indication de la classe à laquelle il appartient dans une classification déterminée.

07.04.05

subscript
A *symbol* that is associated with the name of a *set* to identify a particular *subset* or *element*.

07.04.05

indice
Symbole associé au nom d'un *ensemble* pour identifier un *sous-ensemble* ou un *élément* particulier.

07.04.06

index (in programming)
A *subscript*, of *integer* value, that identifies the position of an *item* of *data* with respect to some other item of data.

07.04.06

indice (de position)
Indice, de valeur entière, qui identifie la position d'une *donnée* par rapport à une autre donnée.

07.04.07

level number
rank
A reference number that indicates the position of an *item* in a hierarchic arrangement.

07.04.07

numéro de niveau
Numéro de référence utilisé pour indiquer la position d'un *article* dans un ensemble hiérarchisé.

07.04.08

reserved word

key word (deprecated in this sense)

A *word* of a *source language* whose meaning is fixed by the particular rules of that language and cannot be altered for the convenience of any one *computer program* expressed in the source language; computer programs expressed in the source language may also be prohibited from using such words in other contexts in the computer program.

Exemples :

- 1 "SIN" may be a reserved word to *call* a *subroutine* for computing the sine function.
- 2 The "COBOL words" such as :
"OCCURS";
"INDEXED BY".

07.04.09

picture

In a *programming language*, a description of a *character string* in which each *position* has associated with it a *symbol* representing the properties of the *character* that may occupy it.

Exemple : In COBOL, 9999 is used as a picture of any 4-digit * numeric word.

07.04.10

literal

figurative constant (deprecated in this sense)

In a *source program*, an explicit representation of the value of an *item*, which value must be unaltered during any translation of the source program.

Exemple : The word "FAIL" in the *instruction* "If x = 0 print FAIL".

07.04.11

filler

padding (deprecated in this sense)

One or more *characters* adjacent to an *item* of *data* that serve to bring its representation up to a specified size.

07.04.12

padding

A technique that incorporates *fillers* in *data*.

07.05 PROGRAM VARIABLES AND PARAMETERS

07.05.01

variable (in programming)

A *character* or a group of characters that refers to a value and, in the *execution* of a *computer program*, corresponds to an *address*.

07.04.08

mot réservé

Mot d'un *langage d'origine* dont la signification et, éventuellement, les conditions d'emploi sont fixées par les règles qui définissent ce *langage*, et qui ne peut pas être utilisé, avec une signification ou dans des conditions différentes, à l'intérieur des *programmes* écrits dans ce *langage*.

Exemples :

- 1 «SIN» peut être un mot réservé pour l'*appel* d'un *sous-programme* de calcul de la fonction sinus.
- 2 Les «mots COBOL» tels que :
«OCCURS»;
«INDEXED BY».

07.04.09

image

Dans un *langage de programmation*, description d'une *chaîne de caractères* dans laquelle à chaque *position* est associé un *symbole* représentant les propriétés du *caractère* qui peut occuper cette position.

Exemple : En COBOL, 9999 est utilisé comme image de tout mot numérique de quatre chiffres.

07.04.10

libellé

(symbole) littéral

Représentation explicite de la valeur d'un *article* qui ne doit être modifiée par aucune traduction du *programme d'origine* où elle figure.

Exemple : Dans l'*instruction* «Si x = 0 imprimer FAUTE», le mot «FAUTE» est un libellé.

07.04.11

élément de remplissage

Caractère ou groupe de caractères servant à donner à un *article* adjacent une taille déterminée.

07.04.12

remplissage

Technique employée pour compléter des *données* à l'aide d'un *élément de remplissage*.

07.05 VARIABLES ET PARAMÈTRES DU PROGRAMME

07.05.01

variable (en programmation)

Caractère ou groupe de caractères qui représente une valeur et qui, lors de l'*exécution* d'un *programme de calculateur*, correspond à une *adresse*.

07.05.02

to bind (of a variable)
to set (of a variable) /GB/

To assign a value to a *variable*; in particular, to assign a value to a *parameter*.

07.05.03

program generated parameter
dynamic parameter

A *parameter* that is *bound* during the *execution* of a *computer program*.

07.05.04

external program parameter
program parameter (deprecated /GB/)

In a *computer program*, a *parameter* that must be *bound* during the *calling* of the *computer program*.

07.05.05

preset parameter

A *parameter* that is *bound* when the *computer program* is constructed, for example when it is *flowcharted*, **coded* or *compiled*.

07.05.06

to preset

To establish an initial condition, such as the control values of a *loop*, or the value to which a *parameter* is to be *bound*.

07.05.07

parameter word

A *word* that directly or indirectly provides or designates one or more *parameters*.

07.05.08

skeletal code

A *set of instructions* in which some parts such as *addresses* must be completed or specified in detail each time the set is used.

07.06 ORGANIZATIONAL PROGRAMS AND TECHNIQUES

07.06.01

supervisory program
executive program
supervisor

A *computer program*, usually part of an *operating system*, that controls the *execution* of other *computer programs* and regulates the flow of work in a *data processing system*.

07.05.02

fixer (une variable)

Attribuer une valeur à une *variable*, notamment à un *paramètre*.

07.05.03

paramètre dynamique

Paramètre * *fixé* au cours de l'*exécution* d'un *programme de calculateur*.

07.05.04

paramètre externe

Paramètre d'un *programme de calculateur*, qui doit être *fixé* lors de l'*appel* de ce *programme*.

07.05.05

paramètre prédéfini

Paramètre que l'on *fixe* lors de l'élaboration d'un *programme de calculateur*, par exemple lors du tracé de l'organigramme, ou bien lors de l'écriture ou de la compilation du *programme*.

07.05.06

prédéfinir

prérégler

Établir une condition initiale, telle que l'ensemble des valeurs de commande d'une *boucle*, ou bien la valeur à laquelle un *paramètre* doit être *fixé*.

07.05.07

mot-paramètre

Mot fournissant ou désignant, directement ou indirectement, un ou plusieurs *paramètres*.

07.05.08

squelette

Ensemble d'*instructions* dont certaines parties, notamment les *adresses*, doivent être complétées ou particularisées à chaque fois que cet ensemble est employé.

07.06 TECHNIQUES ET PROGRAMMES D'ORGANISATION

07.06.01

(programme) superviseur

Programme de calculateur qui commande l'*exécution* d'autres *programmes de calculateur* et l'enchaînement des travaux dans un *système de traitement de l'information*; un tel *programme* fait généralement partie d'un *système d'exploitation*.

07.06.02

monitor (program)
monitoring program

A *computer program* that observes, regulates, controls or verifies the operations of a *data processing system*.

07.06.03

housekeeping operation
overhead operation

An *operation* that facilitates the *execution* of a *computer program* without making a direct contribution.

Exemples : The initialization of *storage* areas; the execution of a *calling sequence*.

07.06.04

table lookup

A *procedure* for obtaining the value corresponding to an *argument* from a *table* of values.

07.06.05

indicator

An *item* of *data* that may be interrogated to determine whether a particular condition has been satisfied in the *execution* of a *computer program*.

Exemples : A *switch indicator*; an *overflow indicator*.

07.07 UTILITY ROUTINES AND TECHNIQUES

07.07.01

utility routine [program]
service routine [program]

A *routine* [A *computer program*] in general support of the processes of a *computer*.

Exemples : An *input routine*; a *diagnostic program*; a *trace program*; a sort program.

NOTE — Computer programs and routines are distinguished in this context by the form and frequency of their use; see 01.04.02 and 01.04.08.

07.07.02

input routine [program]

A *utility routine* [A *utility program*] that organizes the *input process* of a *computer*.

NOTE — Computer programs and routines are distinguished in this context by the form and frequency of their use; see 01.04.02 and 01.04.08.

07.07.03

bootstrap

A set of *instructions* that cause additional instructions to be loaded until the complete *computer program* is in *storage*.

07.06.02

(programme) moniteur

Programme de calculateur qui surveille, organise, commande ou vérifie le fonctionnement d'un *système de traitement de l'information*.

07.06.03

opération d'aménagement

Opération qui facilite l'*exécution* d'un *programme de calculateur* mais n'y contribue pas directement.

Exemples : Écriture de valeurs initiales dans des zones de *mémoire*; exécution d'une *séquence d'appel*.

07.06.04

consultation de table

Procédure permettant d'obtenir, dans une *table* de valeurs, la valeur correspondant à un *argument* donné.

07.06.05

indicateur (en programmation)

Donnée qui peut être consultée pour savoir si l'on a satisfait à une condition déterminée lors de l'*exécution* d'un *programme de calculateur*.

Exemples : *Indicateur d'aiguillage*, *indicateur de dépassement de capacité*.

07.07 TECHNIQUES ET PROGRAMMES DE SERVICE

07.07.01

programme de service
programme utilitaire

Routine destinée à faciliter les traitements effectués par un *calculateur*.

Exemples : *Programme d'entrée*, * *programme de diagnostic*, * *programme de traçage*, *programme de tri*.

07.07.02

programme d'entrée
programme d'introduction

Programme de service destiné à assurer les opérations d'*entrée* dans un *calculateur*.

07.07.03

amorce

Ensemble d'*instructions* servant à charger d'autres instructions jusqu'à ce que le *programme de calculateur* complet soit en *mémoire*.

07.07.04

bootstrap (loader)

An *input *routine* in which preset *computer operations* are used to load a *bootstrap*.

07.07.05

to bootstrap

To use a *bootstrap*.

07.07.06

initial program loader

A *bootstrap loader* used in a *computer* to load that part of an *operating system* needed to load the remainder of the operating system.

07.07.07

output routine [program]

A *utility routine* [A *utility program*] that organizes the *output process* of a *computer*.

NOTE — *Computer programs* and *routines* are distinguished in this context by the form and frequency of their use; see 01.04.02 and 01.04.08.

07.08 SUBROUTINES AND MULTIPLE USE PROGRAMS

07.08.01

subroutine

A *sequenced *set of statements* that may be used in one or more *computer programs* and at one or more points in a *computer program*.

07.08.02

open subroutine

direct insert subroutine

A *subroutine*, one copy of which must be inserted at each place, the subroutine is used in a *computer program*.

07.08.03

closed subroutine

A *subroutine*, one copy of which suffices to be *linked* by *calling sequences* for use at more than one place in a *computer program*.

07.07.04

programme d'amorce

Routine d'entrée utilisant des *opérations de calculateurs* prédéterminées pour charger une *amorce*.

07.07.05

amorcer

Employer une *amorce*.

07.07.06

procédure de chargement initial

programme de chargement initial

Programme d'amorce utilisé par un *calculateur* pour charger la partie du *système d'exploitation* nécessaire au chargement du reste du système d'exploitation.

07.07.07

programme de sortie

Programme de service destiné à organiser les opérations de *sortie* d'un *calculateur*.

07.08 PROGRAMMES À USAGE MULTIPLE ET SOUS-PROGRAMMES

07.08.01

sous-programme

*Ensemble *ordonné d'instructions* pouvant être employé dans un ou plusieurs *programmes de calculateur* ou bien en un ou plusieurs points d'un même programme.

07.08.02

sous-programme ouvert

Sous-programme dont un exemplaire doit être inséré à chacun des emplacements où il est employé dans un *programme de calculateur*.

07.08.03

sous-programme fermé

Sous-programme pour lequel il suffit d'insérer un seul exemplaire pour pouvoir le *relier*, par des *séquences d'appel*, à un ou plusieurs emplacements d'un *programme de calculateur*.

07.08.04**to nest**

To incorporate a structure or structures of some kind into a structure of the same kind.

Exemples : To nest one *loop* (the nested loop) within another loop (the nesting loop); to nest one *subroutine* (the nested subroutine) within another subroutine (the nesting subroutine).

07.08.05**reusable program [routine]**

A computer program [A routine] that may be loaded once and *executed* repeatedly, subject to the requirements that any *instructions* that are modified during its *execution* are returned to their initial states and that its *external program parameters* are preserved unchanged.

07.08.06**reentrant program [routine] [subroutine]
reenterable program [routine] [subroutine]**

A computer program [A routine] [A subroutine] that may be entered repeatedly and may be entered before prior *executions* of the same computer program [routine] [subroutine] have been completed, subject to the requirement that neither its *external program parameters* nor any *instructions* are modified during its execution.

NOTE — A reentrant program, routine or subroutine may be used by more than one computer program simultaneously.

07.08.07**recursive subroutine [routine]**

A subroutine [A routine] that may be used as a subroutine of itself, calling itself directly or being called by another subroutine, one that it itself has called.

NOTE — The use of a recursive subroutine or routine usually requires the keeping of records of the status of its unfinished uses in, for example, a *pushdown list*.

07.09 ENTRY POINTS, EXITS AND LINKS BETWEEN PROGRAMS**07.09.01****entry point**

entrance /USA/

entry /GB/

The *address* or the *label* of the first *instruction* * *executed* upon entering a *computer program*, a *routine* or a *subroutine*.

NOTE — A computer program, a routine or a subroutine may have a number of different entry points, each perhaps corresponding to a different function or purpose.

07.08.04**emboîter**

Incorporer une structure, ou plusieurs structures d'un certain type, à l'intérieur d'une structure de même type.

Exemples : *Boucles* emboîtées, *sous-programmes* emboîtés.

07.08.05**programme [routine] réutilisable**

Programme de calculateur [Routine] que l'on peut charger une seule fois, puis *exécuter* plusieurs fois; ce programme doit mettre dans un état initial déterminé toutes les *instructions* qui sont modifiées lors de l'*exécution* et ne doit pas modifier ses *paramètres externes*.

07.08.06**programme [routine] [sous-programme] reentrant**

Programme de calculateur [Routine] [Sous-programme] dans lequel on peut entrer à plusieurs reprises, éventuellement avant que ses précédentes *exécutions* ne soient terminées; ni ses *instructions*, ni ses *paramètres externes* ne doivent être modifiés pendant son exécution.

NOTE — Un programme [routine] [sous-programme] reentrant peut être employé simultanément par plusieurs programmes.

07.08.07**sous-programme récursif [routine récursive]**

Sous-programme [Routine] qui peut être employé comme son propre sous-programme, soit qu'il s'*appelle* directement, soit qu'il soit appelé par un autre sous-programme auquel lui-même a fait appel.

NOTE — L'emploi d'une routine ou d'un sous-programme récursif exige généralement que l'on garde un enregistrement de l'état des passages non terminés, par exemple dans une *liste inversée*.

07.09 DÉBUTS ET FINS DE PROGRAMMES ET LIENS ENTRE PROGRAMMES**07.09.01****point d'entrée**

adresse d'entrée

Adresse ou *étiquette* de la première *instruction* * *exécutée* lorsqu'on entre dans un *programme de calculateur*, dans un *sous-programme* ou dans une *routine*.

NOTE — Un programme de calculateur, un sous-programme ou une routine peut avoir plusieurs *points d'entrée* distincts correspondant chacun à une fonction différente.

07.09.02

entry conditions

The conditions to be specified on entering a *computer program*, a *routine*, or a *subroutine*.

Example : The *addresses* of those locations from which the program, routine or subroutine will take its *operands* and of those locations with which its *entry points* and *exits* will be *linked*.

07.09.03

to trigger

To cause the immediate *execution* of a *computer program*, often by intervention from the external environment, for example by means of a manually controlled *jump* to an *entry point*.

07.09.04

call

The action of bringing a *computer program*, a *routine*, or a *subroutine* into effect, usually by specifying the *entry conditions* and *jumping* to an *entry point*.

07.09.05

to call

In *computer * programming*, to *execute* a *call*.

07.09.06

calling sequence

An arrangement of *instructions*, and in some cases of *data* also, that is necessary to perform a *call*.

07.09.07

to prestore

To *store*, before a *computer program*, *routine* or *subroutine* is entered, *data* that are required by the computer program, the routine or the subroutine.

07.09.08

exit

Any *instruction* in a *computer program*, in a *routine* or in a *subroutine* after the *execution* of which control is no longer exercised by that computer program, that routine or that subroutine.

07.09.09

link

linkage

In *computer * programming*, a part of a *computer program*, in some cases a single *instruction* or an *address*, that passes control and *parameters* between separate portions of the computer program.

07.09.10

to link

In *computer * programming*, to provide a *link*.

07.09.02

conditions d'entrée

Conditions à déterminer avant de déclencher l'exécution d'un *programme de calculateur*, d'un *sous-programme* ou d'une *routine*.

Exemple : *Adresses* des emplacements où le programme prendra ses *opérandes* et adresses des emplacements auxquels il reliera ses *points d'entrée* et ses *instructions de sortie*.

07.09.03

déclencher

lancer

Provoquer l'*exécution* immédiate d'un *programme de calculateur*, généralement par une intervention du milieu extérieur, par exemple en commandant manuellement un *saut* vers le *point d'entrée*.

07.09.04

appel

Opération déclenchant l'*exécution* d'un *programme de calculateur*, d'un *sous-programme* ou d'une *routine*; cette opération consiste généralement à déterminer les *conditions d'entrée* et à effectuer un *saut* jusqu'à un *point d'entrée*.

07.09.05

appeler

En *programmation*, effectuer un *appel*.

07.09.06

séquence d'appel

Groupe d'*instructions*, et éventuellement de *données*, nécessaire pour effectuer un *appel*.

07.09.07

pré-ranger

Ranger en mémoire les *données* exigées par un *programme de calculateur*, un *sous-programme* ou une *routine*, avant de commencer son *exécution*.

07.09.08

instruction de sortie

Instruction d'un *programme de calculateur*, d'un *sous-programme* ou d'une *routine* après l'*exécution* de laquelle la commande n'appartient plus à ce programme, ce sous-programme ou cette routine.

07.09.09

lien

En *programmation*, partie de *programme de calculateur* qui transfère la commande et transmet les *paramètres* entre deux tronçons de programme distincts; cette partie de programme peut éventuellement se réduire à une seule *instruction* ou à une *adresse*.

07.09.10

relier

En *programmation*, établir un *lien*.

07.09.11

1 to return (transitive)

Within a *subroutine*, to *bind* a *variable* in the *computer program* that *called* the subroutine.

07.09.12

2 to return (intransitive)

Within a *subroutine*, to effect a *link* to the *computer program* that *called* the subroutine.

07.09.13

reentry point

The *address* or the *label* of the *instruction* at which the *computer program* that *called* a *subroutine* is reentered from the subroutine.

07.09.11

renvoyer

Dans un *sous-programme*, * *fixer* une *variable* du *programme de calculateur* qui a *appelé* ce sous-programme.

07.09.12

retourner (verbe intransitif)

Dans un *sous-programme*, mettre en œuvre un *lien* vers le *programme de calculateur* qui a *appelé* ce sous-programme.

07.09.13

point de retour

Adresse ou *étiquette* de l'*instruction* à laquelle, après l'*exécution* d'un *sous-programme*, on reprend le déroulement du *programme* qui avait appelé ce sous-programme.

07.10 LOOPS

07.10.01

loop

A *set of instructions* that may be *executed* repeatedly while a certain condition prevails.

NOTE — In some implementations, no test is made to determine whether the condition prevails until the loop has been executed once.

07.10.02

closed loop

A *loop* that has no *exit* and whose *execution* can be interrupted only by intervention from outside the *computer program* in which the loop is included.

07.10.03

breakpoint halt

dynamic stop /GB/

breakpoint instruction

A *closed loop* consisting of a single *jump instruction* that effects a *jump* to itself, often used to achieve a *breakpoint*.

07.10.04

to unwind

To state explicitly and in full, without the use of *modifiers*, all the *instructions* that are involved in the *execution* of a *loop*.

07.10.05

1 straight line coding

A *set of instructions* in which there are no *loops*.

07.10.06

2 straight line coding

A *programming* technique in which *loops* are avoided by *unwinding*.

07.10 BOUCLES

07.10.01

boucle

Ensemble d'*instructions* dont l'*exécution* peut être répétée tant que certaines conditions sont remplies.

NOTE — Pour certaines applications, on ne vérifie ces conditions qu'après une exécution de la boucle.

07.10.02

boucle fermée

Boucle ne possédant pas d'*instruction de sortie* et dont on ne peut sortir que par une intervention extérieure au *programme de calculateur* contenant cette boucle.

07.10.03

arrêt dynamique

Boucle fermée constituée d'une seule *instruction de saut* qui provoque un *saut* vers elle-même; cette boucle est souvent employée pour mettre en place un *point d'interruption*.

07.10.04

dérouler

Établir clairement et complètement, sans emploi de *modificateur*, toutes les *instructions* mises en jeu dans l'*exécution* d'une *boucle*.

07.10.05

séquence sans boucle

Ensemble d'*instructions* dans lequel il n'y a pas de *boucle*.

07.10.06

programmation sans boucle

Technique de *programmation* dans laquelle les *instructions* sont *déroulées* pour éviter les *boucles*.

07.11 JUMPS AND BRANCHING

07.11.01

jump

branch (deprecated in this sense)

transfer (deprecated in this sense)

In the *execution* of a *computer program*, a departure from the implicit or declared *order* in which *instructions* are being *executed*.

07.11.02

unconditional jump

A *jump* that takes place whenever the *instruction* that specifies it is *executed*.

07.11.03

conditional jump

A *jump* that takes place only when the *instruction* that specifies it is *executed* and specified conditions are satisfied.

07.11.04

to branch

In the *execution* of a *computer program*, to select one from a number of alternative sets of *instructions*.

07.11.05

branchpoint

A point in a *computer program* at which *branching* occurs, in particular the *address* or the *label* of an *instruction*.

07.11.06

switchpoint

switch /USA/

In a *computer program*, a *parameter* that controls *branching* and is *bound* prior to the *branchpoint* being reached.

07.11.07

switch indicator

flag

An *indicator* that determines or shows the setting of a *switchpoint*.

07.11.08

trap

An unprogrammed *conditional jump* to a specified *address* that is automatically activated by *hardware*, a recording being made of the location from which the jump occurred.

07.12 STORAGE ALLOCATIONS AND SEGMENTATION OF PROGRAMS

07.12.01

storage allocation

The assignment of *storage* areas to specified *data*.

07.11 SAUTS ET AIGUILLAGES

07.11.01

saut

Dans l'*exécution* d'un *programme de calculateur*, abandon de l'ordre implicite suivant lequel les *instructions* doivent être *exécutées*.

07.11.02

saut inconditionnel

Saut effectué à chaque *exécution* de l'*instruction* qui le désigne.

07.11.03

saut conditionnel

Saut effectué à chaque *exécution* de l'*instruction* qui le désigne, seulement si des conditions déterminées sont remplies.

07.11.04

brancher

aiguiller

Dans l'*exécution* d'un *programme de calculateur*, choisir un ensemble d'*instructions* parmi plusieurs ensembles possibles.

07.11.05

point de branchement

bifurcation

Point d'un *programme de calculateur*, notamment *adresse* ou *étiquette* d'une *instruction*, où est effectué le choix d'un ensemble d'*instructions* parmi plusieurs ensembles possibles.

07.11.06

aiguillage

Dans un *programme de calculateur*, * *paramètre* qui commande le choix d'un ensemble d'*instructions* parmi plusieurs ensembles possibles; ce paramètre doit être *fixé* avant que l'on n'atteigne le *point de branchement*.

07.11.07

indicateur d'aiguillage

drapeau

Indicateur qui détermine ou met en évidence la position d'un *aiguillage*.

07.11.08

déroutement

Saut conditionnel vers une *adresse* déterminée, réalisé automatiquement par les circuits câblés de la machine, et non par programme; ce saut s'accompagne d'un enregistrement de l'emplacement auquel il est apparu.

07.12 ATTRIBUTION DE MÉMOIRE ET SEGMENTATION DES PROGRAMMES

07.12.01

attribution de mémoire

Assignation de zones de *mémoire* à des *données* déterminées.

07.12.02

dynamic storage allocation

A *storage allocation* technique in which the *storage* areas assigned to *computer programs* and to *data* are determined by criteria applied at the moment of need.

07.12.03

to relocate

To move a *computer program* or part of a computer program, and to adjust the necessary *address* references so that the computer program can be *executed* after being moved.

07.12.04

relocatable program

A *computer program* that is in such a form that it may be *relocated*.

07.12.05

segment

section (deprecated)

chapter (deprecated)

partition (deprecated)

A self-contained portion of a *computer program* that may be *executed* without the entire computer program necessarily being maintained in the *internal storage* at any one time.

07.12.06

to segment

to section (deprecated)

to partition (deprecated)

To divide a *computer program* into *segments*.

07.12.07

overlay

In a *computer program*, a *segment* that is not permanently maintained in *internal storage*.

07.12.08

to overlay

In the *execution* of a *computer program*, to load a *segment* of the computer program in a *storage* area hitherto occupied by parts of the computer program that are not currently needed.

07.13 INTERRUPTIONS AND RESTARTS

07.13.01

breakpoint

A place in a *computer program*, usually specified by an *instruction*, where its *execution* may be *interrupted* by external intervention or by a *monitor program*.

07.12.02

attribution dynamique de mémoire

Technique d'*attribution de mémoire*, selon laquelle les zones de *mémoire* attribuées au *programme de calculateur* et aux *données* sont déterminées en fonction de critères évalués au moment où le besoin se présente.

07.12.03

translater

Déplacer en *mémoire* un *programme de calculateur*, ou un tronçon de programme, en modifiant, si nécessaire, les références aux *adresses*, de sorte que le programme puisse être *exécuté* à son nouvel emplacement.

07.12.04

programme translatable

Programme de calculateur présenté sous une forme telle qu'il puisse être *translaté*.

07.12.05

segment

Tronçon autonome d'un *programme de calculateur*, pouvant être *exécuté* sans qu'il soit nécessaire de maintenir à aucun moment la totalité du programme en *mémoire interne*.

07.12.06

segmenter

Diviser un *programme de calculateur* en *segments*.

07.12.07

segment (de recouvrement)

Dans un *programme de calculateur*, **segment* qui n'est pas maintenu en permanence dans la *mémoire interne*.

07.12.08

recouvrir

Dans l'*exécution* d'un *programme de calculateur*, charger un *segment* du programme dans une zone de *mémoire* occupée jusque là par des tronçons du programme dont on n'a momentanément pas besoin.

07.13 INTERRUPTIONS ET REPRISES

07.13.01

point d'interruption

Point d'un *programme de calculateur*, déterminé généralement au moyen d'une *instruction*, où l'*exécution* de ce programme peut être *interrompue* par une intervention extérieure ou par un *programme moniteur*.

07.13.02
checkpoint

A place in a *computer program* at which a *check* is made or at which a recording of *data* is made for *restart* purposes.

07.13.03
restart

The resumption of the *execution* of a *computer program* using the *data* recorded at a *checkpoint*.

07.13.04
to restart

To perform a *restart*.

07.13.05
restart point

rescue point /GB/

A place in a *computer program* at which its *execution* may be *restarted*; in particular, the *address* of a *restart instruction*.

07.13.06
restart condition

restart point (deprecated in this sense)

In the *execution* of a *computer program*, a condition that can be re-established and that permits a *restart* of the computer program.

07.14 DUMPING

07.14.01
to dump

To *write* the contents of a *storage*, or of part of a storage, usually from an *internal storage* onto an external *medium*, for a specific purpose such as to allow other use of the storage, as a safeguard against *faults* or *errors*, or in connection with *debugging*.

07.14.02
dump (routine)

A *utility routine* that *dumps*.

07.14.03
dump

Data that have been *dumped*.

07.14.04
selective dump

The *dumping* of the contents of one or more specified *storage areas*.

07.14.05
change dump

A *selective dump* from those *storage locations* whose contents have changed.

07.13.02
point de contrôle

Point d'un *programme de calculateur* où l'on effectue un *contrôle* ou bien où l'on enregistre des *données* en vue d'une *reprise* ultérieure.

07.13.03
reprise

Poursuite de l'*exécution* d'un *programme de calculateur*, en employant les *données* enregistrées à un *point de contrôle*.

07.13.04
reprendre

Effectuer la *reprise* d'un *programme de calculateur*.

07.13.05
point de reprise

Point d'un *programme de calculateur*, notamment *adresse* d'une *instruction de reprise*, où l'on peut *reprendre* le déroulement du programme.

07.13.06
condition de reprise

Dans l'*exécution* d'un *programme de calculateur*, condition que l'on doit remplir pour effectuer une *reprise* de ce programme.

07.14 VIDAGES

07.14.01
vider

Écrire le contenu de tout ou partie d'une *mémoire*, généralement une *mémoire interne*, sur un *support externe*, dans un but particulier, tel que libérer l'emplacement initial, se prémunir contre des *fautes* ou des *erreurs* ou procéder à une *mise au point*.

07.14.02
programme de vidage

Programme de service destiné à *vider*.

07.14.03
(résultat de) vidage

Données obtenues par l'*exécution* d'un *programme de vidage*.

07.14.04
vidage sélectif

Vidage portant sur des zones particulières de la *mémoire*.

07.14.05
vidage après changement

Vidage sélectif portant sur les *emplacements de mémoire* dont le contenu a varié depuis le dernier vidage.

07.14.06**dynamic dump**

Dumping performed during the *execution* of a *computer program*, usually under the control of that computer program.

07.14.07**snapshot dump**

A *dynamic dump* of the contents of one more specified *storage areas*.

07.14.08**static dump**

Dumping that is performed at a particular point in time with respect to a machine, often at the end of a *run*, and usually under the control of the *computer operator* or a *supervisory program*.

07.14.09**postmortem dump**

Dumping that is performed at the end of a *run*, usually for purposes of *debugging*, auditing or *documentation*.

07.15 PROGRAM MAINTENANCE**07.15.01****to debug** (in programming)

To detect, to trace and to eliminate *mistakes* in *computer programs* or in other *software*.

NOTE — In the English language, "trouble shoot" and "trouble shooting" may be used in lieu of "debug" and "debugging".

07.15.02**diagnostic program**

A *computer program* that recognizes, locates and explains either a *fault* in equipment or a *mistake* in a computer program.

07.15.03**trace program**

A *computer program* that performs a *check* on another computer program by exhibiting the sequence in which the *instructions* are executed and usually the results of executing them.

07.15.04**snapshot (program)****snapshot trace program** /S/

A *trace program* that produces *output data* only for selected *instructions* or for selected conditions.

07.15.05**checking program**

A *computer program* that examines other computer programs or *sets of data* for *mistakes of syntax*.

07.14.06**vidage dynamique**

Vidage effectué au cours de l'*exécution* d'un *programme de calculateur* et généralement commandé par ce programme.

07.14.07**vidage dynamique sélectif**

Vidage dynamique portant sur des zones particulières de *mémoire*.

07.14.08**vidage statique**

Vidage effectué à un instant particulier repéré dans le déroulement du travail, généralement à la fin d'un *passage*, et généralement commandé par l'opérateur du *calculateur* ou par un *programme superviseur*.

07.14.09**vidage postmortem**

Vidage effectué à la fin d'un *passage* et employé généralement pour *mettre au point* un programme ou pour des contrôles ou dans un but documentaire.

07.15 MAINTENANCE DES PROGRAMMES**07.15.01****mettre au point** (en programmation)

Détecter, localiser et éliminer les *erreurs de programmation*.

07.15.02**programme de diagnostic**

Programme de calculateur destiné à reconnaître, localiser et identifier soit des *défauts* du matériel, soit des *erreurs* dans un programme de calculateur.

07.15.03**programme de traçage****programme de dépistage****programme d'analyse**

Programme de calculateur destiné à effectuer un *contrôle* d'un autre programme de calculateur, en mettant en évidence la succession des *instructions* qui sont *exécutées* et, généralement, leurs résultats.

07.15.04**programme d'analyse sélective****programme d'instantané**

Programme de traçage ne sortant de *données* que pour certaines *instructions* ou dans certaines conditions définies au préalable.

07.15.05**programme de contrôle**

Programme de calculateur destiné à rechercher des *erreurs de syntaxe* dans d'autres programmes de calculateur ou dans des *données*.

07.15.06
to patch
To make an improvised modification.

07.16 INSTRUCTIONS – GENERAL TERMS

07.16.01
computer instruction
machine instruction
An *instruction* that can be recognized by the *central processing unit* of the *computer* for which it is designed.

07.16.02
presumptive instruction
unmodified instruction (deprecated)
An *instruction* that is not an *effective instruction* until it has been modified in a prescribed manner.

07.16.03
effective instruction
actual instruction (deprecated)
An *instruction* that may be *executed* without modification.

07.16.04
absolute instruction
A *computer instruction* in its final, *executable* form.

07.16.05
macroinstruction
macro(instruction)
An *instruction* in a *source language* that is to be replaced by a defined sequence of instructions in the same *source language*.

NOTE — The macroinstruction may also specify values for *parameters* in the instructions that are to replace it.

07.16.06
macrodefinition
macrodeclaration
A *declaration* that provides the *skeletal code* which a *macrogenerator* uses in replacing a *macroinstruction*.

07.16.07
instruction set
The set of the *instructions* of a *computer*, of a *programming language*, or of the programming languages in a *programming system*.

07.16.08
(computer) instruction code
machine code
machine language (deprecated in this sense)
A *code* used to represent the *instructions* in an *instruction set*.

07.16.09
operation code
order code (deprecated)
A *code* used to represent the *operations* of a *computer*.

07.15.06
rapiécer
Apporter une modification de façon sommaire.

07.16 INSTRUCTIONS – TERMES GÉNÉRAUX

07.16.01
instruction-machine
instruction de calculateur
Instruction pouvant être reconnue par l'*unité centrale de traitement* du *calculateur* pour lequel elle est conçue.

07.16.02
instruction primitive
Instruction qui doit subir une modification prédéterminée pour devenir une *instruction effective*.

07.16.03
instruction effective
Instruction qui peut être *exécutée* sans modification.

07.16.04
instruction absolue
Instruction-machine sous sa forme finale exécutable.

07.16.05
macro-instruction
Instruction écrite dans un *langage d'origine* et destinée à être remplacée par une suite déterminée d'instructions dans le même langage d'origine.

NOTE — La macro-instruction peut également spécifier les valeurs qui doivent être attribuées à certains *paramètres* dans les instructions qui la remplaceront.

07.16.06
macro-définition
macro-déclaration
Pseudo-instruction qui fournit le *squelette* employé par un *macro-générateur* pour remplacer une *macro-instruction*.

07.16.07
jeu d'instructions
Ensemble des *instructions* d'un *calculateur* ou d'un *langage de programmation* ou des langages de programmation d'un *système de programmation*.

07.16.08
code des instructions (d'un calculateur)
Code employé pour représenter les *instructions* d'un *jeu d'instructions*.

07.16.09
code d'opérations
Code employé pour représenter les *opérations* d'un *calculateur*.

07.16.10

instruction word

A *word* that represents an *instruction*.

07.16.11

label (in programming)

An *identifier* of an *instruction*.

07.16.12

(instruction) modifier

A *word* or a part of a *word* that is used to alter an *instruction*.

07.16.13

microinstruction

An *instruction* of a *microprogram*.

07.17 INSTRUCTION FORMATS

07.17.01

instruction format

The layout of an *instruction* showing its constituent parts.

07.17.02

operation part

operator part

function part

A part of an *instruction* that usually contains only an explicit specification of the *operation* to be performed.

NOTE — For an exception to “usually”, see *implied addressing*.

07.17.03

address part

A part of an *instruction* that usually contains only an *address* or part of an *address*.

NOTE — For an exception to “usually”, see *immediate address*.

07.17.04

multiaddress instruction

multiple address instruction

An *instruction* that contains more than one *address part*.

07.17.05

one [two] ... [n] address instruction

An *instruction* that contains one [two] ... [n] *address parts*.

07.17.06

one-plus-one [two-plus-one] ... [n-plus-one] address instruction

An *instruction* that contains two [three] ... [n + 1] *address parts*, the “plus-one” *address* being that of the *instruction* that is to be *executed* next unless otherwise specified.

07.16.10

mot instruction

Mot qui représente une *instruction*.

07.16.11

étiquette (en programmation)

Identificateur d'une *instruction*.

07.16.12

modificateur (d'instruction)

Mot, ou partie de *mot*, employé pour modifier une *instruction*.

07.16.13

micro-instruction

Instruction d'un *microprogramme*.

07.17 CLICHÉS D'INSTRUCTION

07.17.01

cliché d'instruction

modèle d'instruction

forme d'instruction

Disposition d'une *instruction* mettant en évidence ses différentes parties.

07.17.02

partie type d'opération

Partie d'une *instruction* qui, en général, contient seulement l'indication explicite de l'*opération* à effectuer.

NOTE — La partie type d'opération peut, éventuellement, contenir d'autres renseignements, par exemple dans le cas d'un *adressage implicite*.

07.17.03

partie adresse

zone adresse

Partie d'une *instruction* qui, en général, contient seulement tout ou partie d'une *adresse*.

NOTE — La partie adresse peut contenir, au lieu d'une *adresse*, d'autres informations, par exemple un *opérande immédiat*.

07.17.04

instruction à plusieurs adresses

Instruction qui contient plusieurs *parties adresses*.

07.17.05

instruction à une [deux] ... [n] adresse

Instruction contenant une [deux] ... [n] *partie adresse*.

07.17.06

instruction à une plus une [à deux plus une] ... [à n plus une] adresses

Instruction contenant deux [trois] ... [n + 1] *parties adresses*, l'*adresse* «plus un» étant celle de l'*instruction* qui, sauf spécification contraire, est la prochaine à *exécuter*.

07.17.07

zero address instruction

An *instruction* that contains no *address part*, being used when the *address* is implicit or when no address is required.

07.18 TYPES OF INSTRUCTION

07.18.01

immediate instruction

An *instruction* that contains within itself an *operand* for the *operation* specified, rather than an *address* of the operand.

07.18.02

direct instruction

An *instruction* that contains the *direct address* of an *operand* for the *operation* specified.

07.18.03

indirect instruction

An *instruction* that contains the *indirect address* of an *operand* for the *operation* specified.

07.18.04

arithmetic instruction

arithmetical instruction

An *instruction* in which the *operation part* specifies an *arithmetic operation*.

07.18.05

logic instruction

An *instruction* in which the *operation part* specifies a *logic operation*.

07.18.06

discrimination instruction

decision instruction

An *instruction* of the class of instructions that comprises *branch instructions* and *conditional jump instructions*.

07.18.07

branch instruction

decision instruction /USA/

An *instruction* that controls *branching*.

07.18.08

jump instruction

branch instruction (deprecated in this sense)

control transfer instruction (deprecated)

transfer instruction (deprecated)

An *instruction* that specifies a *jump*.

07.18.09

unconditional jump instruction

unconditional branch instruction (deprecated)

unconditional control transfer instruction (deprecated)

unconditional transfer instruction (deprecated)

An *instruction* that specifies an *unconditional jump*.

07.17.07

instruction sans adresse

Instruction ne contenant pas de *partie adresse*, et employée soit parce que l'*adresse* n'est pas nécessaire, soit parce qu'elle est implicite.

07.18 TYPES D'INSTRUCTIONS

07.18.01

instruction à opérande immédiat

instruction à adresse immédiate

Instruction qui contient en elle-même un *opérande* de l'*opération*, et non l'*adresse* de cet opérande.

07.18.02

instruction à adresse directe

Instruction qui contient l'*adresse directe* d'un *opérande* de l'*opération* spécifiée par cette instruction.

07.18.03

instruction à adresse indirecte

Instruction qui contient l'*adresse indirecte* d'un *opérande* de l'*opération* spécifiée par cette instruction.

07.18.04

instruction arithmétique

Instruction dont la *partie type d'opération* spécifie une *opération arithmétique*.

07.18.05

instruction logique

Instruction dont la *partie type d'opération* spécifie une *opération logique*.

07.18.06

instruction de décision

Instruction de branchement ou *instruction de saut conditionnel*.

07.18.07

instruction de branchement

Instruction effectuant le choix d'un ensemble d'instructions parmi plusieurs ensembles possibles.

07.18.08

instruction de saut

instruction de rupture de séquence

Instruction qui spécifie un *saut*.

07.18.09

instruction de saut inconditionnel

Instruction qui spécifie un *saut inconditionnel*.

07.18.10

conditional jump instruction

conditional branch instruction (deprecated)

conditional control transfer instruction (deprecated)

conditional transfer instruction (deprecated)

An *instruction* that specifies a *conditional jump* and the conditions that have to be satisfied for the conditional jump to occur.

07.18.11

stop instruction

An *exit* that specifies the termination of the *execution* of a *computer program*.

07.18.12

pause instruction

halt instruction

An *instruction* that specifies the suspension of the *execution* of a *computer program*.

NOTE — A pause instruction is usually not an *exit*.

07.18.13

optional pause instruction

optional stop instruction

An *instruction* that allows manual suspension of the *execution* of the *computer program*.

07.18.14

restart instruction

An *instruction* in a *computer program* at which the computer program may be *restarted*.

07.18.15

no operation instruction

NO OP (abbreviation)

do nothing operation

An *instruction* whose *execution* causes a *computer* to do nothing other than to proceed to the next instruction to be executed.

07.18.16

dummy instruction

An *item* of *data* in the form of an *instruction* that is inserted in a *set* of instructions, but is not intended to be *executed*.

07.18.17

privileged instruction

An *instruction* that may only be used by a *supervisory program*.

07.19 ADDRESSES

07.19.01

instruction address

The *address* of an *instruction word*.

07.18.10

instruction de saut conditionnel

Instruction qui spécifie un *saut conditionnel*, en indiquant les conditions selon lesquelles ce saut est effectué.

07.18.11

instruction d'arrêt

Instruction de sortie qui indique la fin de l'*exécution* d'un *programme de calculateur*.

07.18.12

instruction de pause

Instruction qui suspend l'*exécution* d'un *programme de calculateur*.

NOTE — Une instruction de pause n'est généralement pas une *instruction de sortie*.

07.18.13

instruction d'arrêt facultatif

Instruction qui suspend l'*exécution* du *programme de calculateur* à condition qu'une commande manuelle l'ait préalablement autorisé.

07.18.14

instruction de reprise

Instruction à partir de laquelle on peut *reprendre* le déroulement d'un *programme de calculateur*.

07.18.15

instruction inefficace

Instruction qui donne ordre au *calculateur* de ne rien faire d'autre que de traiter l'*instruction* qui suit immédiatement.

07.18.16

instruction fictive

Donnée ayant la forme d'une *instruction* et insérée dans une suite d'instructions, mais qui n'est pas destinée à être *exécutée*.

07.18.17

instruction privilégiée

Instruction qui ne peut être employée que par un *programme superviseur*.

07.19 ADRESSES

07.19.01

adresse d'instruction

Adresse d'un *mot instruction*.

07.19.02 effective address The contents of the <i>address part</i> of an <i>effective instruction</i> .	07.19.02 adresse effective Contenu de la <i>partie adresse</i> d'une <i>instruction effective</i> .
07.19.03 absolute address machine address (deprecated in this sense) An address in a <i>computer language</i> that identifies a <i>storage location</i> or a device without the use of any intermediate reference.	07.19.03 adresse absolue adresse machine (terme déconseillé dans ce sens) <i>Adresse</i> qui, en <i>langage-machine</i> , identifie directement, sans référence intermédiaire, un <i>emplacement de mémoire</i> ou un dispositif.
07.19.04 symbolic address An <i>address</i> expressed in a form convenient for <i>computer* programming</i> .	07.19.04 adresse symbolique <i>Adresse</i> exprimée sous une forme symbolique commode pour la <i>programmation</i> .
07.19.05 base address A <i>numeric</i> value that is used as a reference in the calculation of <i>addresses</i> in the <i>execution</i> of a <i>computer program</i> .	07.19.05 adresse de base adresse base Valeur <i>numérique</i> employée comme repère dans les calculs d' <i>adresses</i> lors de l'exécution d'un <i>programme de calculateur</i> .
07.19.06 relative address An <i>address</i> expressed as a difference with respect to a <i>base address</i> .	07.19.06 adresse relative <i>Adresse</i> exprimée sous forme de l'écart avec une <i>adresse de base</i> .
07.19.07 self-relative address A <i>relative address</i> that uses as <i>base address</i> the <i>address</i> of the <i>instruction</i> in which it appears.	07.19.07 adresse auto-relative <i>Adresse relative</i> qui emploie comme <i>adresse de base</i> l' <i>adresse</i> de l' <i>instruction</i> où elle figure.
07.19.08 relocatable address An <i>address</i> that is adjusted when the <i>computer program</i> containing it is <i>relocated</i> .	07.19.08 adresse translatable <i>Adresse</i> destinée à être modifiée lorsque le <i>programme de calculateur</i> qui la contient sera <i>translaté</i> .
07.19.09 immediate address zero-level address The contents of an <i>address part</i> that contains the value of an <i>operand</i> rather than an <i>address</i> .	07.19.09 opérande immédiat adresse immédiate Contenu d'une <i>partie adresse</i> , qui représente la valeur d'un <i>opérande</i> et non une <i>adresse</i> .
07.19.10 direct address one-level address An <i>address</i> that designates the <i>storage location</i> of an <i>item of data</i> to be treated as an <i>operand</i> .	07.19.10 adresse directe <i>Adresse</i> qui désigne l' <i>emplacement de mémoire</i> d'une <i>donnée</i> destinée à être traitée comme un <i>opérande</i> .
07.19.11 indirect address multilevel address An <i>address</i> that designates the <i>storage location</i> of an <i>item of data</i> to be treated as the address of an <i>operand</i> , but not necessarily as its <i>direct address</i> .	07.19.11 adresse indirecte <i>Adresse</i> qui désigne l' <i>emplacement de mémoire</i> d'une <i>donnée</i> destinée à être traitée comme adresse d'un <i>opérande</i> , mais non nécessairement comme son <i>adresse directe</i> .

07.19.12

generated address

synthetic address

An *address* that has been formed as a *result* during the *execution* of a *computer program*.

07.19.12

adresse calculée

Résultat obtenu durant l'exécution d'un programme de calculateur et employé comme *adresse*.

07.20 ADDRESSING METHODS

07.20 MÉTHODES D'ADRESSAGE

07.20.01

to address

To refer to a device or an *item* of *data* by its *address*.

07.20.01

adresser

Désigner une *donnée* ou un dispositif en donnant son *adresse*.

07.20.02

absolute [symbolic] [relative] [self-relative] [immediate] [direct] [indirect] addressing

A method of *addressing* in which the *address part* of an *instruction* contains an *absolute address* [a *symbolic address*] [a *relative address*] [a *self-relative address*] [an *immediate address*] [a *direct address*] [an *indirect address*].

07.20.02

adressage absolu [symbolique] [relatif] [auto-relatif] [immédiat] [direct] [indirect]

Type d'*adressage* dans lequel la *partie adresse* d'une *instruction* contient une *adresse absolue* [une *adresse symbolique*] [une *adresse relative*] [une *adresse auto-relative*] [un *opérande immédiat*] [une *adresse directe*] [une *adresse indirecte*].

07.20.03

deferred addressing

A method of *addressing* in which one *indirect address* is replaced by another to which it refers a predetermined number of times or until the process is terminated by an *indicator*.

07.20.03

adressage différé

Type d'*adressage* dans lequel une *adresse indirecte* renvoie à une autre *adresse indirecte*, ce processus se répétant soit un nombre prédéterminé de fois, soit jusqu'à ce qu'un *indicateur* y mette fin.

07.20.04

implied addressing

A method of *addressing* in which the *operation part* of an *instruction* implicitly addresses *operands*.

07.20.04

adressage implicite

Type d'*adressage* dans lequel la *partie type d'opération* d'une *instruction* * *adresse* implicitement des *opérandes*.

07.20.05

repetitive addressing

A method of *implied addressing*, applicable only to *zero address instructions*, in which the *operation part* of an *instruction* implicitly addresses the *operands* of the last instruction *executed*.

07.20.05

adressage répétitif

Type d'*adressage implicite*, applicable seulement à des *instructions sans adresse*, dans lequel la *partie type d'opération* d'une *instruction* * *adresse* implicitement les *opérandes* de la dernière instruction *exécutée*.

07.20.06

one-ahead addressing

A method of *implied addressing* in which the *operation part* of an *instruction* implicitly addresses the *operands* in the location following the location of the *operands* of the last instruction *executed*.

07.20.06

adressage à progression automatique

Type d'*adressage implicite* dans lequel la *partie type d'opération* d'une *instruction* * *adresse* implicitement les *opérandes* situés immédiatement après les *opérandes* de la dernière instruction *exécutée*.

07.20.07

directory

A *table* of *identifiers* and references to the corresponding *items* of *data*.

07.20.07

répertoire

Table contenant des *identificateurs* et des références correspondant à des *données*.

07.20.08

index word

An *instruction modifier* applied to the *address part* of an *instruction*.

07.20.08

mot d'index

Modificateur appliqué à la *partie adresse* d'une *instruction*.

4 ALPHABETICAL INDEX

A			B		
absolute	absolute address	07.19.03	base	base address	07.19.05
	absolute addressing	07.20.02	bind	to bind	07.05.02
	absolute instruction	07.16.04	bootstrap	bootstrap	07.07.03
access	access-time	12		bootstrap (loader)	07.07.04
actual	actual instruction (<i>deprecated</i>)	07.16.03		to bootstrap	07.07.05
address	absolute address	07.19.03	branch	branch instruction	07.18.07
	address	07.01.11		branch instruction (<i>deprecated</i>)	07.18.08
	address part	07.17.03		branch (<i>deprecated</i>)	07.11.01
	to address	07.20.01		to branch	07.11.04
	base address	07.19.05		conditional branch instruction	
	direct address	07.19.10		(<i>deprecated</i>)	07.18.10
	effective address	07.19.02		unconditional branch instruction	
	generated address	07.19.12		(<i>deprecated</i>)	07.18.09
	immediate address	07.19.09	branchpoint	branchpoint	07.11.05
	indirect address	07.19.11	breakpoint	breakpoint	07.13.01
	instruction address	07.19.01		breakpoint halt	07.10.03
	machine address (<i>deprecated</i>)	07.19.03		breakpoint instruction	07.10.03
	multilevel address	07.19.11	C		
	multiple address instruction	07.17.04			
	n-address instruction	07.17.05			
	n-plus-one address instruction	07.17.06			
	one-address instruction	07.17.05			
	one-level address	07.19.10			
	one-plus-one address instruction	07.17.06			
	relative address	07.19.06			
	relocatable address	07.19.08			
	self-relative address	07.19.07			
	symbolic address	07.19.04	call	call	07.09.04
	synthetic address	07.19.12		to call	07.09.05
	three-address instruction	07.17.05	calling	calling sequence	07.09.06
	three-plus-one address instruction	07.17.06	central	central processing unit	01
	two-address instruction	07.17.05	change	change dump	07.14.05
	two-plus-one address instruction	07.17.06	chapter	chapter (<i>deprecated</i>)	07.12.05
	zero-address instruction	07.17.07	character	character	04
	zero-level address	07.19.09		character string	04
addressing	absolute addressing	07.20.02	check	check	...
	deferred addressing	07.20.03	checking	checking program	07.15.05
	direct addressing	07.20.02	checkpoint	checkpoint	07.13.02
	immediate addressing	07.20.02	closed	closed loop	07.10.02
	implied addressing	07.20.04		closed subroutine	07.08.03
	indirect addressing	07.20.02	code	code	04
	one-ahead addressing	07.20.06		to code	04
	relative addressing	07.20.02		(computer) instruction code	07.16.08
	repetitive addressing	07.20.05		machine code	07.16.08
	self-relative addressing	07.20.02		operation code	07.16.09
	symbolic addressing	07.20.02		order code (<i>deprecated</i>)	07.16.09
	one-ahead addressing	07.20.06		skeletal code	07.05.08
ahead	algorithm	01	coding	1 straight-line coding	07.10.05
algorithm	algorithmic language	07.02.12		2 straight-line coding	07.10.06
algorithmic	dynamic storage allocation	07.12.02	command	command (<i>deprecated</i>)	07.01.09
allocation	storage allocation	07.12.01	comment	comment	07.01.08
	(computer program) annotation	07.01.08	compile	compile duration	07.03.13
annotation	argument	02		compile phase	07.03.12
argument	arithmetic instruction	07.18.04	compiler	to compile	07.03.06
arithmetic	arithmetic operation	02		compiler	07.03.07
	arithmetic instruction	07.18.04		compiler generator	07.03.10
arithmetical	artificial language	07.02.03	compiling	compiling phase	07.03.12
artificial	assemble duration	07.03.13		compiling program	07.03.07
assemble	to assemble	07.03.04		compiling time	07.03.13
	assembler	07.03.05	computer	computer	01
assembler	assembling phase	07.03.12		computer instruction	07.16.01
assembling	assembling time	07.03.13		(computer) instruction code	07.16.08
	assembly language	07.02.16		computer language	07.02.15
assembly	assembly phase	07.03.12		computer language (<i>deprecated</i>)	07.02.14
	assembly program	07.03.05		computer operation	10
automatic	automatic data processing	01		computer-oriented language	07.02.14
				computer program	01
				(computer program) annotation	07.01.08
			condition	restart condition	07.13.06
			conditional	conditional branch instruction	
				(<i>deprecated</i>)	07.18.10

conditional	conditional control transfer instruction (<i>deprecated</i>)	07.18.10
	conditional jump	07.11.03
	conditional jump instruction	07.18.10
	conditional transfer instruction (<i>deprecated</i>)	07.18.10
conditions	entry conditions	07.09.02
constant	figurative constant	07.04.03
	figurative constant (<i>deprecated</i>)	07.04.10
control	conditional control transfer instruction (<i>deprecated</i>)	07.18.10
	control transfer instruction (<i>deprecated</i>)	07.18.08
	unconditional control transfer instruction (<i>deprecated</i>)	07.18.09

D

data	automatic data processing	01
	data	01
	data name	07.04.02
	(data processing) system	01
	output data	06
debug	to debug	07.15.01
decision	decision instruction	07.18.06
	decision instruction /USA/	07.18.07
declaration	declaration	07.01.10
	macrodeclaration	07.16.06
	declarative (<i>deprecated</i>)	07.01.10
deferred	deferred addressing	07.20.03
definition	macrodefinition	07.16.06
delay	minimum delay programming	07.01.12
diagnostic	diagnostic program	07.15.02
digit	digit	04
direct	direct address	07.19.10
	direct addressing	07.20.02
	direct insert subroutine	07.08.02
	direct instruction	07.18.02
	directive	07.01.10
	directory	07.20.07
	discrimination instruction	07.18.06
	do nothing operation	07.18.15
	documentation	04
	dummy instruction	07.18.16
directive	change dump	07.14.05
	dump	07.14.03
	dump (routine)	07.14.02
	to dump	07.14.01
	dynamic dump	07.14.06
	postmortem dump	07.14.09
	selective dump	07.14.04
	snapshot dump	07.14.07
	static dump	07.14.08
	assemble duration	07.03.13
duration	compile duration	07.03.13
	run duration	07.03.13
	translate duration	07.03.13
	dynamic dump	07.14.06
dynamic	dynamic parameter	07.05.03
	dynamic stop /GB/	07.10.03
	dynamic storage allocation	07.12.02

E

effective	effective address	07.19.02
	effective instruction	07.16.03
element	element	02
entrance	entrance /USA/	07.09.01
entry	entry /GB/	07.09.01
	entry conditions	07.09.02
	entry point	07.09.01
	error	02
error	error	02
execute	to execute	10

execute	execute phase	07.03.12
execution	execution	10
executing	executing phase	07.03.12
executive	executive program	07.06.01
exit	exit	07.09.08
external	external program parameter	07.05.04

F

fault	fault	14
figurative	figurative constant	07.04.03
	figurative constant (<i>deprecated</i>)	07.04.10
filler	filler	07.04.11
flag	flag	07.11.07
flowchart	flowchart	01
format	instruction format	07.17.01
function	function part	07.17.02

G

generated	generated address	07.19.12
	program-generated parameter	07.05.03
generating	macro-generating program	07.03.11
generator	compiler generator	07.03.10

H

halt	breakpoint halt	07.10.03
	halt instruction	07.18.12
high	high-level language	07.02.17
housekeeping	housekeeping operation	07.06.03

I

identifier	identifier	07.04.01
immediate	immediate address	07.19.09
	immediate addressing	07.20.02
	immediate instruction	07.18.01
	implied addressing	07.20.04
implied	implied addressing	07.20.04
index	index	07.04.06
	index word	07.20.08
indicator	indicator	07.06.05
	switch indicator	07.11.07
indirect	indirect address	07.19.11
	indirect addressing	07.20.02
	indirect instruction	07.18.03
	information	01
information	information retrieval	01
initial	initial program loader	07.07.06
input	input process	06
	input program	07.07.02
	input routine	07.07.02
	direct insert subroutine	07.08.02
insert instruction	absolute instruction	07.16.04
	actual instruction (<i>deprecated</i>)	07.16.03
	arithmetic instruction	07.18.04
	arithmetical instruction	07.18.04
	branch instruction	07.18.07
	branch instruction (<i>deprecated</i>)	07.18.08
	breakpoint instruction	07.10.03
	computer instruction	07.16.01
	(computer) instruction code	07.16.08
	conditional branch instruction (<i>deprecated</i>)	07.18.10
	conditional control transfer instruction (<i>deprecated</i>)	01.18.10
	conditional jump instruction (<i>deprecated</i>)	07.18.10