Programmation des microprocesseurs d'architecture x86-64 sous Linux

Patrick Cégielski
Konstantin Verchinine

Mai 2019

Legal Notice

Université Paris Est - Créteil – IUT
Route forestière Hurtault
F-77300 Fontainebleau
cegielski@u-pec.fr

Préface

Notre but, dans ce livre, est d'initier le lecteur à la programmation des microprocesseurs 64 bits d'*Intel* et AMD, d'architecture dite x86-64, implémentée sur presque tous les microprocesseurs actuels, mais qui demeure encore sous-employée.

Il y a trois façons d'utiliser les instructions (fines) d'un microprocesseur :

— en programmant en langage machine, le moyen le plus sûr mais le plus rustre ;

— en programmant en langage d'assemblage et en laissant à l'assembleur le soin de traduire le programme source en un programme en langage machine ;

— en insérant une partie en langage d'assemblage dans un programme en langage C (ou tout autre langage évolué).

Pourquoi utiliser les instructions du microprocesseur et ne pas se contenter du langage C, qui reste cependant le langage de référence ?

— pour augmenter la performance de certains programmes, comme nous le verrons à propos du calcul du produit scalaire de deux vecteurs de grande dimension, en particulier en éliminant certaines redondances par rapport au code obtenu grâce à un compilateur à partir d'un langage de haut niveau ;

— pour apporter de nouvelles fonctionnalités, en particulier en accédant à des registres non pris en compte par les appels système existants, comme nous le verrons à propos de `TSC` (*Time Stamp Counter*), le compteur de tops d'horloge ;

— pour comprendre certains problèmes de sécurité, comme nous le verrons à propos de la pile (*stack overflow*).

Un inconvénient est que le programme n'est plus portable d'un microprocesseur à un autre, voire d'un système d'exploitation à un autre (si on utilise des appels système).

Chapitre 1

Introduction

1.1 Les microprocesseurs 64 bits

1.1.1 Définition

L'histoire des *microprocesseurs* est fortement liée à *Intel*, entreprise née en 1968 afin d'exploiter une technologie venant alors d'apparaître : les circuits intégrés. Elle débute comme *fondeur*, c'est-à-dire fabricant de circuits intégrés pas nécessairement conçus par elle, de la première application des circuits intégrés à connaître un succès certain : les mémoires à semi-conducteurs, appelées à remplacer les mémoires à tores de ferrite employées alors. Elle en devient très rapidement le numéro un. Le microprocesseur est inventé au sein de la société en 1971, dont elle demeurera le numéro un sur ce marché durant quarante ans.

Le premier microprocesseur, dénommé `4004`, est un microprocesseur dit 4 bits, dans la mesure où les entrées-sorties ainsi que les opérations (addition et multiplication) s'effectuent sur 4 bits en une seule instruction. Il a été conçu afin dêtre incorporé dans une nouvelle gamme de calculatrices de bureaux, pour lesquelles un tel microprocesseur est bien adapté : 4 bits permettent de manipuler les chiffres de '0' à '9'. Par contre, un tel microprocesseur ne peut pas servir de cœur d'un ordinateur, aussi petit soit-il. *Intel* sort ensuite le microprocesseur 8 bits *8008*, bientôt suivi des *8080* et *8085*. Les microprocesseurs 8 bits permettent l'émergence des micro-ordinateurs à usage domestique, mais *Intel* perd à ce moment-là la première place du marché au profit de concurrents. La société s'attelle alors au microprocesseur 16 bits *8086*, qui va connaître un énorme succès en permettant le déploiement des micro-ordinateurs à une très grande échelle dans les entreprises.

Sur sa lancée, *Intel* « innove » pour une question de prix de revient : elle crée le *8088*, permettant des entrées-sorties sur 8 bits mais des calculs sur 16 bits (le « cœur » du nouveau microprocesseur est le même que le *8086*). IBM choisit ce microprocesseur pour son PC, le micro-ordinateur adopté par les entreprises, destiné jusque-là surtout à un public d'« amateurs » ou pour des tâches spécialisées. Le nombre de bits n'est plus une caractéristique très précise de la puissance du microproceseur. *Intel* reprenant la première place des fondeurs, conçoit des microprocesseurs 32 bits (`i386`, `i486` puis *pentium*) qui lui permet de détenir la première place

sur le marché des micro-ordinateurs et crée le *mode protégé*, destiné à faciliter la conception des systèmes d'exploitation.

Il n'est plus alors facile de savoir ce qu'il faut entendre par *microprocesseur n bits* : le nombre de broches des données, le nombre de broches des adresses, la taille des registres, la taille des opérandes sur laquelle est effectuée une opération en peu de cycles machine ?

Définition.- *On parle de* **microprocesseur** n **bits** *lorsque ses registres généraux ont une taille de* n *bits.*

En 1982, *Intel* accorde à AMD (*Advanced Micro Devices*) une licence pour produire les microprocesseurs `8086` et `8088`, afin de renforcer la position de son architecture sur le marché. À la suite d'une bataille juridique, AMD obtient en 1995 le droit de produire des microprocesseurs fondés sur l'architecture IA-32, conçue par *Intel*. AMD conçoit en 2003 l'architecture 64 bits `AMD64`, totalement compatible avec l'IA-32, ce qui lui permet d'atteindre une part de près de 25 % sur le marché des microprocesseurs `x86`. Cette architecture est adoptée par *Intel* quelques années plus tard sous le nom de `EM64T` (*Extended Memory 64 bits Technology*).

Définition.- *On appelle* `x86-64`*, au lieu des dénominations* `AMD64` *ou* `EM64T` *propres aux fondeurs concernés, l'extension du jeu d'instructions* `x86` *pour une architecture 64 bits. Cette extension permet la gestion des entiers sur 64 bits, avec pour corollaire un adressage mémoire allant bien au-delà de la limite des 4 GiO du* `x86`*. À cela s'ajoute le doublement (de 8 à 16) du nombre de registres généralistes.*

1.1.2 Utilisation des microprocesseurs 64 bits

L'utilisation principale des microprocesseurs 64 bits est l'adressage de la mémoire : 32 bits ne peuvent normalement pas adresser plus de 4 Gio (2^{32} octets) de mémoire centrale, tandis que les processeurs 64 bits peuvent en adresser 16 Eio (2^{64} octets). C'est pourquoi dès qu'il y a plus de 4 Gio de RAM sur une machine, la mémoire située au-delà de ce seuil ne sera (directement) adressable qu'en mode 64 bits.

Une autre application serait d'effectuer les opérations sur 64 bits en peu de cycles machine de façon à améliorer la performance des calculs. Mais cela n'est pas le cas de l'architecture `x86-64`. *Intel* a essayé de le faire avec le microprocesseur *Itanium* (architecture **x64**), mais celle-ci n'a jamais connu de succès et a été abandonné en janvier 2019.

1.2 Aide matérielle au temps partagé

1.2.1 L'emploi des ordinateurs en temps partagé (1962)

Maurice WILKES, le concepteur du premier ordinateur opérationnel, écrit un petit livre [Wil-68] dans lequel il fait le point sur l'*emploi partagé des ordinateurs*, qui a commencé à être mis en place à partir de 1962 et est alors encore peu développé.

Il rappelle que, sur les premiers ordinateurs, l'utilisateur passait le temps qu'il voulait à tester son programme mais qu'« *on ne fut pas long à réaliser que c'était là une bien mauvaise façon d'employer une machine rare et coûteuse.* » On est donc passé au **traitement par lots** (*batch processing* en anglais) : « *Les travaux sont chargés par lots sur ruban magnétique, souvent à l'aide d'un petit ordinateur auxiliaire. Chaque lot de travaux est placé sur l'ordinateur principal ; les résultats sortent sur un autre ruban magnétique et sont ensuite imprimés.* [...] *Ces développements ont sans aucun doute amélioré le rendement des ordinateurs ; ils ont eu cependant l'effet regrettable d'éloigner l'utilisateur de l'ordinateur et de diminuer, en particulier dans le cas de programmes en développement, la rapidité du travail de mise au point.* » On est donc passé ensuite au **temps partagé** (*time-sharing* en anglais) : « *Il est maintenant possible aux utilisateurs d'être reliés par une paire de câbles à un ordinateur puissant qui peut être situé à proximité ou parfois à des kilomètres de distance. Tous les utilisateurs, quels qu'ils soient, ont accès instantanément à l'ordinateur et peuvent obtenir une réponse à leurs demandes sous la réserve que l'ordinateur doit partager son temps entre tous les utilisateurs. Le développement de tels systèmes est, cependant, encore en enfance.* »

1.2.2 Programmation des applications et programmation système

Maurice WILKES continue en distinguant ce que nous appelons maintenant **programmation des applications** et **programmation système** : « *En décrivant un système d'emploi partagé, une distinction doit être faite entre les programmes des utilisateurs et les programmes qui assurent diverses fonctions d'administration ou de commutation. Ces derniers sont plus ou moins interconnectés et désignés collectivement sous le nom de* superviseur. [...] *Au lieu de programme d'utilisateur, il est souvent préférable d'employer le terme programme-objet ; ceci inclut des programmes qui, bien que n'appartenant pas à un utilisateur donné, sont traités par le système exactement de la même manière que les programmes d'utilisateurs. L'une des plus importantes fonctions du superviseur est déviter que les programmes-objets interfèrent entre eux. Dans ce but, le superviseur doit avoir certains privilèges qui sont refusés aux programmes-objets. Parmi ceux-ci, on peut citer le contrôle des dispositifs d'entrée et de sortie, l'asservissement des interruptions, ainsi que l'établissement des limites de mémoire dans lesquelles les programmes-objets peuvent opérer. Beaucoup d'ordinateurs fonctionnent suivant deux modes opératoires, un mode ordinaire pour le déroulement du programme-objet, un mode privilégié réservé au superviseur ; quelques types d'instructions peuvent être exécutés seulement en mode privilégié.* »

Les concepts sont là, même si le vocabulaire a changé : on parle de *système d'exploitation*, *programme d'application* et *programme système* au lieu de *superviseur*, *programme-objet* et programme assurant diverses fonctions d'administration et de commutation. Un programme d'application est donc, par conséquent, un programme conçu pour un système d'exploitation donné, ce dernier étant en place. La programmation système concerne la programmation du système d'exploitation (y compris les pilotes de périphériques).

1.2.3 Aide matérielle au temps partagé

Au moment où Maurice WILKES écrit, en 1968, l'emploi partagé des ordinateurs repose entièrement sur du logiciel. Le but de son livre est d'inciter à répartir cette fonctionnalité entre logiciel et matériel.

Mais il va falloir un certain temps avant qu'il ne soit entendu.

Intel introduit le **mode protégé** dans ses microprocesseurs à partir du `80286` pour mettre en place la partie matérielle réclamée par Maurice WILKES. Nous aborderons donc la programmation des applications dans la première partie, réservant la programmation système à une seconde partie.

Comme le dit Maurice WILKES, la mise en place de l'emploi partagé des ordinateurs repose à la fois sur une partie matérielle (le mode protégé pour les microprocesseurs *Intel*) mais aussi sur une partie logicielle, une partie du système d'exploitation de nos jours. La programmation des applications ne peut donc pas faire complètement fi du système d'exploitation utilisé. Tous nos exemples concerneront *Linux*, pour une architecture x86-64.

1.3 Bibliographie

[Wil-68] WILKES, Maurice Vincent, **Time-sharing computer systems**, MacDonald and Co., London, 1968. Tr. fr. **Emploi partagé des ordinateurs**, Dunod, 1971, III + 124 p.