

Jamil Alioui

Réinventer l'utilisation

[10'207 signes]

Contribution au magazine CiNN, été 2017.

La distinction entre logiciel et matériel, inaugurée par la représentation symbolique d'états matériels sur des supports amovibles – et donc interchangeables et indépendants de la machine –, a individué en même temps la programmation comme activité humaine et le programme comme unité textuelle, produit de cette activité. Πρόγραμμα, *programma*, de « devant » et « écriture », signifie initialement « affiche », « ordre du jour », et donc d'une certaine manière : « écriture publique ». En outre, la première machine programmée fut un métier à tisser, aux alentours de 1730, information intéressante lorsqu'on sait que le mot « texte » trouve son origine dans le verbe latin « *texere* » qui signifie, justement, « tisser ». L'émergence de langages de programmation dits de haut niveau a permis d'écrire des programmes dans des langages toujours plus proches du langage ordinaire (programmation dite « orientée objets »), mais ces programmes doivent être compilés pour pouvoir être exécutés par l'ordinateur, *i.e.* correspondre à sa structure matérielle. Or, l'opération de compilation est une opération cryptogène difficilement réversible qui complique la décompilation du produit dans un langage aisément lisible. De malicieux petits commerçants ont alors exploité ce fait, de concert avec les vendeurs de matériel, et en ont profité pour spécialiser les métiers informatiques et séparer ainsi les descendants du scribe en deux ensembles : d'un côté [la secrétaire](#) a perdu son genre féminin pour devenir l'utilisateur générique d'aujourd'hui et, de l'autre, le programmeur a asservi l'utilisateur en rédigeant des textes fondateurs pourtant toujours déjà obsolètes. Ainsi est né l'utilisateur lambda.

Le programme n'est pourtant rien de plus qu'une description d'un état matériel de la machine qui permet d'effectuer d'autres écritures, par ex. sur l'écran. Le calcul n'est donc pas l'opération fondamentale d'un ordinateur. Ce sont le *stockage*, le *chargement* et l'*exécution* de textes qui constituent ce que l'ordinateur *fait* au minimum derrière l'écran sur lequel les résultats de l'exécution – eux-mêmes textuels – se dévoilent. L'ordinateur, par sa nature intrinsèquement symbolique, est donc *éventuellement* une machine à calculer là où, *de toute façon*, il est [une machine d'écriture](#).

Or, si l'on considère l'ordinateur comme une machine d'écriture, l'écriture de programmes constitue aujourd'hui le seul et unique moyen d'en libérer la *pleine* puissance. C'est que l'ordinateur est une machine d'écriture non pas au sens où le logiciel Word (par exemple) permet d'« écrire sur ordinateur » – ce qui ne veut pas dire grand-chose – mais au sens où Word est lui-même un logiciel écrit dont le texte demeure secrètement conservé dans les archives scellées de Microsoft. *La condition de possibilité de l'utilisateur apparaît alors comme le renoncement à un*

pouvoir d'invention et, par conséquent, comme l'abandon d'une puissance. L'utilisateur d'une plume Parker serait bien contrarié d'apprendre qu'en dépensant une somme d'argent importante à la papeterie il acquiert non pas la plume en question mais uniquement le droit temporaire de l'utiliser, et qu'il devra bientôt dépenser la même somme – sinon davantage – pour pouvoir continuer d'écrire sous prétexte que le papier n'est plus compatible avec la plume. Pourtant ce sont bien des « licences d'utilisation » qui, la majeure partie du temps, sont « accordées » aux utilisateurs contre des sommes d'argent parfois invraisemblables. Ces sommes ont été si élevées par le passé que certaines firmes se trouvent aujourd'hui en mesure de se présenter sur le marché comme des [acteurs de l'émancipation sociale](#) sous le prétexte très en vogue de la gratuité. En fait, les dispositifs gratuits ne sont la plupart du temps conçus que pour accroître la dépendance des utilisateurs qui, en acceptant la « mise à jour gratuite » qu'on leur « propose automatiquement » (c'est-à-dire qu'on leur impose), ne font que fructifier un marché qui ne doit son existence qu'à un malentendu sur le concept d'écriture et sur sa portée (cf. la paradigmatique [réification des émojis](#) opérée par Apple qui n'est pas sans rappeler celle, perpétuelle, du novlangue d'Orwell) ; ceci, sans parler de la publicité et du trafic d'informations qui font désormais de tous les utilisateurs des « utilisés », comme l'a efficacement remarqué [Richard Stallman](#).

Au moins deux figures sont nées en réaction à tout ceci : le *power-user* et le *hacker*. Le *power-user* ne programme pas, mais il se débrouille assez bien pour être nettement plus autonome que l'utilisateur lambda. Son tempérament individualiste lui confère un intérêt pragmatique pour l'informatique : malin, il a compris très tôt qu'elle lui permet d'être plus efficace dans son emploi. Le *power-user* demeure pourtant un utilisateur qui perpétue des conditions d'aliénation collective, les adjudicateurs s'en accommodent donc volontiers, même, ils en profitent en lui reléguant une partie du support qu'ils devraient eux-mêmes fournir en l'encourageant à contribuer à des forums semi-publics. En un mot : parce qu'il ne considère pas l'ordinateur comme une machine d'écriture, le *power-user* ne constitue pas une figure de résistance. Le *hacker*, par contre, possède une compétence de *reverse engineering* qui confirme la nature grammatotechnique des ordinateurs dont les vendeurs souhaiteraient l'oubli. Le *hacker* arrive effectivement à décrypter un programme qui a été compilé en vue de le modifier (par ex. pour en *cracker* les fonction d'authentification de licence), il met par conséquent en péril l'idée selon laquelle le devenir-exécutable d'un texte serait une opération irréversible et rappelle au monde que l'informationnel est un projet public et que la sécurité informatique est une affaire politique. Le *hacker* constitue donc une intéressante figure de résistance par le fait qu'il utilise l'ordinateur comme le scientifique utilise l'écriture : de la façon la plus exhaustive et puissante qui soit. Pirater, n'est-ce pas par ailleurs réinventer l'utilisation ?

Un monde informationnel est un monde où la technique constitue le moyen d'un dessein culturel, d'un projet collectif, où l'infrastructure favorise d'abord l'échange, le partage, le transit et les flux. Le cran d'arrêt qui existe encore ici et là entre la production d'un bien symbolique et sa réception tend à disparaître au profit d'un flux d'information continu – un texte –, alimenté et consommé par tous les acteurs – humains et non-humains – du processus culturel, comme une action constitutive à grande échelle. L'encyclopédie est un bon exemple de projet informationnel, de même que les bibliothèques, où l'information est collectée et mise à disposition de façon à former un potentiel générique de connaissances, pour l'ensemble des êtres. La bibliothèque est aussi un concept

informatique : dans sa version non-anthropologique, elle consiste en un fragment de code que d'autres programmes peuvent utiliser pour l'exécution d'une fonction particulière. Une architecture ouverte, comme Linux, exige que les bibliothèques soient délivrées dans deux versions : l'une exécutable par l'ordinateur (et illisible par le programmeur), l'autre non-exécutable mais lisible et modifiable par le programmeur. Une architecture fermée, comme Windows, ne met les bibliothèques à disposition que dans une version compilée, c'est-à-dire cryptée, que l'on ne peut qu'*utiliser* sans les modifier. En optant pour une telle solution, les grandes firmes aliènent l'idée de programme comme écrit public ; d'ailleurs le mot « programme » n'est plus à la mode, il a subrepticement été remplacé, d'abord par celui de « logiciel » puis, plus récemment, par celui d'« application ».

Nous vivons dans un monde qui voudrait [devenir informationnel](#) mais qui n'est encore qu'informatique pour la seule raison que les ordinateurs ne sont pas conçus comme les acteurs d'un projet collectif mais comme un moyen de faire toujours plus de profit (par conséquent ils deviennent une fin en soi). Dans le monde dans lequel nous vivons, il est normal qu'un chercheur traite des données décisives à sa contribution scientifique avec un logiciel qu'il n'a pas programmé lui-même. Dans certains cas, les chercheurs ne savent même pas installer de tels logiciels : IBM SPSS, Microsoft Word, Adobe Photoshop, Apple Keynote, des logiciels que l'on ne pourra jamais tutoyer en ceci qu'ils possèdent aussi un nom de famille, un titre de noblesse, un privilège. L'utilisation d'un logiciel de ce type est donc une opération *magique* contraire aux exigences méthodologiques de la science, qui demanderait plutôt que cette opération soit *technique*, et par conséquent inventive.

Problématiser l'informatique c'est donc la reconnecter technologiquement ([et non simplement techniquement](#)) à un projet de culture informationnelle qui refuse en même temps la soumission de l'utilisateur et celle de l'ordinateur au profit d'un processus d'invention collective. Sans un tel projet, il est parfaitement normal que les « problèmes informatiques » ne dépassent pas les « problèmes d'utilisation » et confèrent au support informatique un rôle de poseur de sparadraps : il n'y a de « problèmes informatiques » que parce que l'informatique n'est pas problématisée.

Remettre intégralement la responsabilité aux multinationales perpétue une attitude passive ; ne sont-ce pas vous et moi qui avons laissé ces privés « prendre en charge » ce qui constitue pourtant les conditions de possibilité d'un projet public ? Ne serait-il pas temps de reprendre la main ? ... par exemple en boycottant Microsoft Word et en installant Linux et LibreOffice sur nos machines ? ... en contribuant à l'enrichissement de Wikipedia plutôt qu'à celui de Facebook ? ... en privilégiant une relation technique aux machines, dans une logique d'invention, plutôt que l'obtention magique de résultats immédiats, dans une logique d'utilisation ?

Il est certain que si nous décidons d'apprendre à lire et à écrire les textes qui constituent ces programmes qui accompagnent nos vies comme nous avons appris à lire et à écrire des articles scientifiques, il y aura de la décroissance. Mais l'alphabétisation de tous ne devrait-elle pas primer sur le profit de quelques uns ?