Microsoft veut remplacer tout son code C/C++ par du Rust d’ici 2030

Modern C

Galen Hunt, l’un des principaux ingénieurs de Microsoft, a publié une offre d’annonce détonante : l’entreprise recherche un ingénieur pour aider à la transition intégrale du code C/C++ vers Rust, qui doit être achevée en à peine cinq ans.

Vincent Hermann
Le 23 décembre à 11h57

6 min Logiciel

Microsoft n’a jamais caché son intérêt pour le Rust. Il a été question un temps d’attendre que l’outillage s’adapte et soit plus mature, mais la version 24H2 de Windows 11 a été la première à introduire du code Rust dans son noyau. Signe clair que la situation avait largement évolué. En février 2025, Paul Thurrott rapportait que la consigne avait été donnée en interne de ne commencer aucun nouveau projet en C ou C++, seulement en Rust.

Le langage, créé initialement par Mozilla, est depuis longtemps géré par une fondation indépendante. Microsoft en était d’ailleurs l’un des principaux membres fondateurs. Le Rust est observé de près par de nombreuses entreprises, particulièrement pour tout ce qui touche à la programmation système. On en trouve d’ailleurs dans le noyau Linux, bien que cette intégration ne se soit pas faite sans heurts. Comme nous l’expliquait récemment Sylvestre Ledru de Mozilla, Firefox en intègre également plusieurs millions de lignes de code, tout comme Chrome.

Sylvestre Ledru (Mozilla) : de Firefox au noyau Linux, la fulgurante ascension du Rust
Logiciel 04/11/2025 10h43 8

Mais Microsoft vient de donner un sérieux coup d’accélérateur : la firme veut remplacer tout son code C/C++ d’ici 2030.

Un projet titanesque

L’annonce n’a pas fait l’objet d’un billet ou d’un communiqué de presse. Elle est présente dans une offre d’emploi publiée par Galen Hunt, l’un des plus anciens ingénieurs logiciels de l’entreprise. L’offre est pour un ingénieur logiciel principal, en présentiel à Redmond.

Elle est cependant vite évacuée au profit d’une déclaration fracassante : « *Mon objectif est d’éliminer toutes les lignes de C et C++ de Microsoft d’ici 2030* ». Galen Hunt indique que la stratégie consiste à mêler IA et algorithmes, et que « *l’étoile polaire* » est d’atteindre « *1 ingénieur, 1 mois, 1 million de lignes de code* ». La tâche est décrite comme « *inimaginable jusqu’ici* ».

L’infrastructure algorithmique de l’entreprise est utilisée actuellement pour créer « *un graphe évolutif sur le code source à grande échelle* ». Après quoi, des agents IA, « *guidés par des algorithmes* », effectuent les modifications, également à grande échelle. Galen Hunt assure que le « *cœur de cette infrastructure fonctionne déjà à grande échelle sur des problèmes tels que la compréhension du code* ».

Une expérience en programmation système de qualité production est exigée. Galen Hunt enchaine sur d’autres paramètres de l’offre et un descriptif de l’équipe travaillant sur ce projet.

Le Rust, toujours le Rust

Plusieurs personnes sont venues témoigner de leur étonnement dans les commentaires. Sur le choix du Rust par exemple : pourquoi ne pas avoir choisi C#, qui présente lui aussi certaines caractéristiques intéressantes pour la sécurité ?

Galen Hunt a répondu : C# est « memory safe », mais pas « concurrent safe ». Comprendre que si C# permet d’éliminer certaines classes de failles de sécurité, notamment via un typage fort, Rust va plus loin. Il est jugé plus adapté à la programmation concurrente, quand plusieurs threads, processus ou tâches évoluent en parallèle, avec ou sans zone mémoire commune. Autre raison, attendue : les performances. Rust fonctionne sans ramasse-miettes (garbage collector) et permet d’atteindre les performances du C++.

L’ingénieur évalue à un milliard le nombre de lignes de code concernées chez Microsoft. Pourquoi un projet aussi démesuré ? Pourquoi ne pas garder le code C/C++ ? « *Pas de sécurité mémoire. Pas de sécurité sur la concurrence. Bien sûr, pour une seule base de code C ou C++, ces qualités peuvent être atteintes par une discipline et un effort extraordinaires – et disparaître en une seule erreur. Avec Rust, cela peut être prouvé par le compilateur* », répond Galen Hunt.

L’annonce a été accueillie avec une certaine incrédulité… y compris dans les rangs mêmes de Microsoft. Rupo Zhang, l’un des responsables de l’ingénierie logicielle de l’entreprise, demande en commentaire sur LinkedIn : « *Vous êtes sérieux ?* ». La question est restée sans réponse.

Relecture critique

Le projet est en effet pharaonique. « *Notre mission est de développer des capacités permettant à Microsoft et à nos clients d’éliminer la dette technique à grande échelle* », indiquait Galen Hunt dans l’annonce. Ce qui implique non seulement la conversion de centaines de millions de lignes de code, mais également les nombreux tests devant être réalisés pour en vérifier la fiabilité et les performances.

L’annonce laisse d’ailleurs entendre que le projet est double : convertir tout le code en Rust et finaliser l’infrastructure capable d’accomplir cette opération. Cette dernière impliquerait notamment que l’intégralité du code de Windows serait convertie en Rust, tout en maintenant la rétrocompatibilité, qui est l’une des marques de fabrique de la plateforme. Début septembre, on apprenait notamment que Microsoft voulait encourager le développement de pilotes en Rust, mais que seules les premières briques de l’infrastructure étaient proposées.

Quoi qu’il en soit, Microsoft répète continuellement depuis plus de dix ans que 70 % des failles de sécurité corrigées sont liées à une mauvaise gestion de la mémoire. Le Rust, bien qu’il élimine pratiquement tous les types de failles dans ce contexte, n’est pas non plus une protection absolue contre toutes les menaces. Il faut encore que le code ait été bien écrit. Comme nous le disait récemment l’ingénieur Horacio Gonzalez (Clever Cloud), la relecture critique a toutes les chances de devenir une compétence très recherchée.

Commentaires (21)

Abonnez-vous pour prendre part au débat Se connecter

fdorin
Aujourd'hui à 12h10

> « 1 ingénieur, 1 mois, 1 million de lignes de code »

C'est juste infaisable. Même avec de l'IA. Ou alors c'est du code d'IA sans aucune relecture derrière...

Le dev, c'est mon métier depuis 15 ans, et ma passion depuis plus de 30. On ne change pas pour le plaisir de changer. Réécrire des portions de code sur lesquels ont est en train d'apporter des modifications, pourquoi pas. Migrer le reste de code quand il reste genre 1% de C pour 99% de Rust, pourquoi pas.

Migrer un OS complet avec une échéance à 5 ans, alors même que le processus de mise à jour n'a fait que se dégrader depuis de nombreux mois, c'est une connerie à tous les niveaux.

Je ne voudrais pas avoir une machine avec ce "nouveau" Windows. Ce sera une hécatombe.

Microsoft semble reprendre l'exemple de Musk quand il était à la tête du DOGE. N'avait-il pas promis de virer tout le COBOL des applications en l'espace de 6 mois ? Je serais curieux de voir où en est ce projet !

👍 20

TexMex
Aujourd'hui à 13h39

C'est parce que ce n'est pas un projet mais un coup marketing.

Oui et puis
Avec des classes remplies de milliers de d'attributs on y arrive vite au million. Il faut au moins 334 000...
"*x lignes de code*" générées par un IDE pour des Get/Set... pas insurmontable. Alors avec une IA bavarde...

👍 1

fdorin
Aujourd'hui à 13h52

> Avec des classes remplies de milliers de d'attributs on y arrive vite au million. Il faut au moins 334 000... "x lignes de code" générées par un IDE pour des Get/Set... pas insurmontable. Alors avec une IA bavarde...

Si on prend un ingénieur, qui bosse 30j par mois (oui, j'ai vu large), 10h par j (toujours large), cela fait 55 attributs par minutes. Même avec un bon IDE, c'est compliqué, 55 attributs par minutes.

Même avec une IA, c'est juste du bullshit.

TexMex
Aujourd'hui à 14h07

Bin... Eclipse / menu '*source*' / "*generate getters & Setters*"... par exemple.

Il y a même des outils pour créer des classes directement à partir de fichiers de propriété dans un tas de langages différent. souvent le cas de support multilingue et '*localisation*' (I18N, etc.) Et même si l'outil n'existe pas on peut le faire tellement le schéma est connu et plutôt simple.

Mon propos n'est pas la productivité humaine. Plutôt sur l'annonce qui semble "*whoaou*" mais en fait; avec les bonnes compétences/méthodes/outils c'est déjà moins brillant.

D'ailleurs il est dit que c'est clairement un chantier de migration de code. Ce qui va se terminer par un outil de conversion automatique. Donc pas un super ingénieur qui code avec des mains à 1000 doigts.

D’où le fait que je dis que c'est du marketing d'abord. Et puis.. Microsoft (et leurs affiliés)... Faut-il détailler comment ils maîtrisent la com ???

fdorin
Aujourd'hui à 15h19

> Il y a même des outils pour créer des classes directement à partir de fichiers de propriété dans un tas de langages différent. souvent le cas de support multilingue et 'localisation' (I18N, etc.) Et même si l'outil n'existe pas on peut le faire tellement le schéma est connu et plutôt simple.

Si tu pars du principe qu'il faut compter la migration de code qui est généré à la base, c'est de la triche

> D'ailleurs il est dit que c'est clairement un chantier de migration de code. Ce qui va se terminer par un outil de conversion automatique. Donc pas un super ingénieur qui code avec des mains à 1000 doigts.

Convertir du code dans 2 langages très proche niveau paradigme (comme le C# et le Java par exemple), c'est déjà un bazar sans nom. Alors convertir 2 langages (C et C++) vers un seul (Rust) aux paradigmes proches mais différents (polymorphisme, héritage, gestion de la mémoire, etc.) je ne suis même pas curieux de voir ce que cela va donner

Et si c'est pour avoir du Rust unsafe partout, l'argument avancé pour la réécriture tombe à l'eau.

> D’où le fait que je dis que c'est du marketing d'abord. Et puis.. Microsoft (et leurs affiliés)... Faut-il détailler comment ils maîtrisent la com ???

Je pense aussi que c'est du marketing. Ca n'empêche pas mon côté technique d'avoir envie de gerber à cette lecture

👍 2

treizarque
Aujourd'hui à 12h15

Courage aux utilisateurs de Windows ! Pardon, je voulais dire aux bêta testeurs.

👍 10 🤣 6

Neliger
Aujourd'hui à 12h19

"Qu'est-ce qui pourrait mal se passer ?" 😹

🤣 6

FLOZz
Aujourd'hui à 14h48

Mais rien voyons ! C'est Microsoft, ils savent ce qu'ils font !

dylem29
Aujourd'hui à 12h27

😥

ilink
Aujourd'hui à 13h10

Déjà si MS faisait un Windows 11 qui fonctionne sans bug et latence..

Que cet ingénieur remette la barre d’adresse d’explorer comme sous Windows 10.

La version 11 est à chier a croire qu’il n’utilise pas son produit !

👍 7

janvi
Modifié le 23/12/2025 à 13h43

Depuis toujours et encore en 2025, des problèmes avec le spouleur d'impression. A votre avis si il le redéveloppe en rust, ca changera ?

🤣 6

guimoploup
Aujourd'hui à 14h10

Nous savons très bien comment se finira ce projet : en 2030 il y aura du C++ ET du RUST.

🥳 4

hypo
Modifié le 23/12/2025 à 14h27

Sauf si d'ici là, on sort quelque chose d'encore "plus à la mode" que Rust, suffit de faire confiance au service marketing pour nous pondre un Rust#

😭 1

BlackLightning
Aujourd'hui à 18h00

Comme quasiment tout les gros projets Rust. Jamais du 100% Rust si on regarde les dépendances il y a toujours un peu de C/C++ qui traîne (au hasard les threads en Rust qui utilise pthreads (c'est pas safe ça) ou encore tokio qui s'appuie sur la libevent)...

pamputt
Aujourd'hui à 16h05

J'ai pas compris pourquoi Microsoft embauche encore des développeurs. Copilot ne sait-il pas faire la migration tout seul ?

😜 5 🤣 8 🥳 1

yl
Aujourd'hui à 18h29

"« Vous êtes sérieux ? ». La question est restée sans réponse."

En réalité, la réponse est dans la question!

L'aspect positif n°1, c'est de pas avoir envisage C#, y'en a qui en ont de bonnes dans cette boite de tordus... Mais c'est un projet qui va tout déstabiliser c'est certain, surtout considérant le foutoir de plus de 4 décennies jamais vraiment remis au propre chez Microsoft! La dette technique qu'ils prétendent résoudre en mode largement automatique, ça va secouer...

L'aspect positif n°2, c'est que cela va tellement les occuper/épuiser qu'ils n'auront guère le temps de polluer l'open-source en faisant "cancer à l'envers".

mvst
Aujourd'hui à 18h48

On sent bien que les gens chez Microsoft sont maintenant objectivés sur l'utilisation de l'IA et sur le nombre de lignes de code, qui est un indicateur universellement reconnu comme pourri, mais le seul indicateur disponible sur le travail des agents IA.
Toutes les métriques disponibles pour Copilot vont dans ce sens : nombre de lignes ajoutées, supprimées proposées, acceptées...

Ici leur objectif est de faire 1 million de lignes de code en un mois et une personne. Pourquoi 1 million ? Pourquoi passer à Rust ?
Je sais bien qu'il y a des arguments déjà annoncés par MS sur leur intérêt pour Rust, mais pourtant la "north star" n'est pas l'amélioration de la performance ou de la stabilité, ni la maintenabilité. Non c'est l'utilisation de l'IA et le nombre de lignes de code.

👍 1

Bylon
Aujourd'hui à 18h50

C'est vrai ! C'est quoi le dernier truc qu'ils ont open-sourcé ? FAT32 ? Youpiiii.

slash622
Aujourd'hui à 19h22

Je suis d'accord avec toi! ce projet c'est juste de la pub deguise.

TheKillerOfComputer
Modifié le 23/12/2025 à 21h18

C'est du même niveau de l'affirmation précédente d'un des leurs selon quoi l'usage de la souris et du clavier semblera aussi étrange en 2030 que de demander à un Gen Z d'utiliser MS-DOS.

Quitte à sortir des âneries, autant que j'y contribue. Le noyau de Linux c'est 40 M de lignes de code en incluant le kernel en soi et différents drivers. Si je me sers de cela comme référence pour Windows tout en extrapolant à la grosse louche pour inclure DirectX, l'interface, Hyper-V, l'explorateur, etc. et le fait que Microsoft externalise les tests vers les usagers histoire que le projet profite des 48 mois restants avant 2030, cela ferait quoi... 10 ingénieurs affectés dessus ? Allez, 20, vu qu'ils ont les moyens ?

psikobare
Aujourd'hui à 23h41

> L’annonce n’a pas fait l’objet d’un billet ou d’un communiqué de presse. Elle est présente dans une offre d’emploi publiée par Galen Hunt, l’un des plus anciens ingénieurs logiciels de l’entreprise. L’offre est pour un ingénieur logiciel principal, en présentiel à Redmond.

Ce n'est en fait pas du tout une annonce. Pareil pour la "déclaration" de Hunt.