



Forum: Les logiciels payants en giveaway

Topic: Macrorit Data Wiper Pro gratuit

Subject: Re: Macrorit Data Wiper Pro gratuit

Publié par: Tof81

Contribution le : 14/10/2016 18:58:28

Citation :

danee a écrit:

Attention !!! les disques SSD ...

Les SSD utilisent une couche d'indirection entre les adresses logiques utilisées par les systèmes informatiques pour accéder aux données et les adresses internes qui identifient stockage physique. Cette couche d'indirection améliore les performances et la fiabilité des SSD en se cachant derrière l'interfaces idiosyncrasique et gère la durée de vie limitée de la mémoire flash.

pour cette raison, il est inutile et contreproductive de vouloir faire une défragmentation sur un SSD ou un **effacement sécurisé** (wipe, schredde, etc...) ou un **effacement des traces dans l'espace libre**, etc... vu que la réécriture des données, ne se fait pas à l'emplacement du fichier original.

le système utilise la fonctionnalité **TRIM** pour mettre le SSD dans un état approprié ... et cette fonctionnalité détruit toutes les données non alloué qui s'y trouvent ...

synonyme: idiosyncrasique = particulier

Salut Danee,

j'ai un peu potassé la littérature et il me semble qu'Indirection et De-indirection sont des applications linéaires bijectives dans un espace vectoriel comme disent les matheux, inventés pour améliorer les performances comme tu le soulignes.

Là où j'ai du mal à te suivre c'est sur le coté improductif de vouloir faire un effacement sécurité vu que l'application est bijective.

Pour la défragmentation, comme je l'ai déjà signalé, il est de plus en plus conseillé d'en faire une tous les 1 à 3 mois, suivant l'usage du SSD. Sur une durée d'utilisation de 5 à 10 ans cela aura aucun effet néfaste sur la durée de vie des cellules

Tof

The basic idea is simple. Let us imagine a system with two levels of mapping, and thus excess indirection. The first indirection F maps items in the A space to items in the B space: $F(A_i) = B_j$. The second indirection G maps items in the B space to those in the C space:

$G(B_j) \rightsquigarrow C_k$. To look up item i , one performs the following “excessive” indirection: $G(F(i))$.

De-indirection removes the second level of indirection by evaluating the second mapping $G()$ for all values mapped by $F()$: $i : F(i) \rightsquigarrow G(F(i))$. Thus, the top-level mapping simply extracts the needed values from the lower level indirection and installs them directly ...