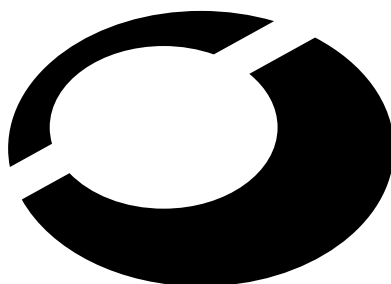


# Le logiciel libre : un défi aux éditeurs propriétaires

## Livre blanc



**Alcôve**

<http://www.alcove.com/>

**Christophe Le Bars, Alcôve  
Septembre 2001**

153, Boulevard Anatole France  
93200 Saint Denis  
France

Tel: +33 1 49 22 68 00

Fax: +33 1 49 22 68 01

*Avec les livres blancs d'Alcôve, bénéficiez de l'expérience de la première société européenne d'expertise sur les logiciels libres.*

**Copyright © 2001 par Alcôve**

**All rights reserved (tous droits réservés).**

## Résumé

Toutes les entreprises doivent aujourd'hui prendre en compte la percée incontestable des logiciels libres tel que Linux ou Apache sur le marché informatique. Ce livre blanc précise les enjeux de ce phénomène en explorant les différences entre ces nouveaux logiciels et ceux traditionnellement produits par les éditeurs propriétaires.

Version 0.8

Référence : WP20010901

## Table des matières

Introduction.....	4
Liberté.....	5
Ouverture .....	6
Qualité.....	7
Sécurité .....	9
Pérennité .....	10
Interopérabilité .....	12
Conclusion .....	12
Glossaire .....	13

## Introduction

Si vous êtes le directeur des systèmes d'informations d'une grande entreprise, alors vous ne le savez peut-être pas, mais les logiciels libres sont déjà chez vous, nombreux et depuis longtemps!

Il y a bien sûr quelques grands projets, visibles, qui utilisent le noyau Linux, le logiciel libre le plus connu aujourd'hui, mais il y a de bonnes chances pour que beaucoup d'autres logiciels libres soient déjà dans la place :

- Vous avez réservé un ou plusieurs noms de domaines sur Internet : sachez que presque tous les serveurs qui vont gérer ce nom de domaine, chez vous ou chez votre fournisseur de services Internet, le feront avec le logiciel Bind;
- Si certaines de vos équipes de développement travaillent en langage C sans demander l'achat ou le renouvellement de licence d'un compilateur C : il est probable qu'elles utilisent maintenant le compilateur GCC et la suite d'utilitaires associée (make, etc);
- Votre entreprise publie directement ou par l'intermédiaire de votre fournisseur d'accès sur Internet un ou plusieurs sites web : vous avez donc plus d'une chance sur deux de faire partie des 60% de sites web qui utilisent le serveur web Apache<sup>2</sup>;
- Comme presque toutes les entreprises, votre système de messagerie est relié à Internet : renseignez-vous pour connaître le nom du serveur SMTP qui gère l'échange d'email avec Internet : il y a de bonnes chances que vous entendiez prononcer l'un des noms suivants : Sendmail, Exim ou Postfix.

Bind, GCC, Apache, Sendmail, Exim ou Postfix, autant de logiciels plus ou moins connus qui ont comme caractéristique commune d'être des logiciels libres, comme le noyau Linux. Beaucoup d'entreprises découvrent aujourd'hui la place déjà incontournable que prennent ces logiciels dans leur système d'information. Même Microsoft qui se défendait de ne jamais utiliser de logiciels libres, a dû admettre récemment qu'il employait FreeBSD - un logiciel libre - dans sa filiale Hotmail, 3 ans après le rachat de cette société. Certaines fonctionnalités et l'efficacité de FreeBSD nécessaires au bon fonctionnement du service Hotmail ne pouvaient simplement pas trouver d'équivalent dans les logiciels propriétaires édités par la firme de Redmond<sup>3</sup>.

En définitive, l'utilisation des logiciels libres par les entreprises est souvent sous-estimée pour plusieurs raisons.

- Le téléchargement, l'installation ou l'usage de ces logiciels n'étant pas soumis à une facturation ou même à une déclaration, ils échappent aux recensements classiques des logiciels propriétaires (comptage des licences vendues) et aux inventaires des services achats de beaucoup d'entreprises.
- Lorsqu'un « serveur Linux » est recensé dans l'entreprise, peu de gens savent qu'en fait il s'agit de nombreuses applications autour du noyau Linux qui ont été installées sur le serveur. Elles sont intégrées dans ce que l'on appelle communément une « distribution » (sous entendu de logiciels libres) dont Linux ne constitue que le coeur, l'interface directe avec le matériel.
- Les logiciels libres étant par nature facilement intégrables, modifiables et personnalisables, ils sont souvent fondus dans une architecture ou une solution globale, définie en tant que telle fonctionnellement, sans identification individuelle de ses composants.

## Liberté

Une drôle d'expression à priori que d'accoler l'adjectif « libre » au mot « logiciel »! En fait, on parle de logiciel libre parce que celui-ci ouvre pour l'entreprise utilisatrice des espaces de liberté supplémentaires et un confort d'utilisation sans commune mesure avec le logiciel propriétaire.

La première liberté offerte est celle d'utiliser le logiciel sans coût de licence proportionnel au nombre de postes installés, d'utilisateurs, de puissance processeur, etc. On imagine bien sûr sans peine les possibilités d'économie d'échelle pour les entreprises en terme d'achats de licence dont l'impact sur les budgets informatiques est incontrôlable. S'affranchir des politiques tarifaires disproportionnées ou erratiques des éditeurs - Oracle et Computer Associates sont coutumiers du fait - est déjà très appréciable, mais n'oublions pas la baisse du coût d'utilisation résultant de l'absence totale de dénombrement et de gestion des licences elle-même.

La seconde liberté est de disposer du code source et de pouvoir le modifier afin de personnaliser le logiciel, d'y apporter de nouvelles fonctionnalités ou de corriger des *bogues*. Les possibilités qu'ouvre cette liberté sont fondamentales, nous verrons par la suite qu'elle porte en elle toute la supériorité du logiciel libre dans son mode de développement.

La troisième liberté est enfin de pouvoir distribuer le logiciel, avec ou sans modification, sans limitation (clients, fournisseurs, partenaires, etc). C'est un aspect nécessaire d'un point de vue économique puisqu'il permet le développement d'un marché autour du logiciel libre. Tous les utilisateurs

n'ayant pas la compétence ou le temps nécessaire pour apporter par eux-mêmes les modifications souhaitées à un logiciel libre, il est indispensable de pouvoir faire appel à un prestataire.

Le marché qui se développe autour du logiciel libre est aujourd'hui en pleine croissance, 28% par an selon IDC, 4 milliards de dollars à l'horizon de 2003 selon Hambrecht. Toujours selon IDC, 9% des investissements informatiques des entreprises seront consacrés à Linux et aux logiciels libres en 2002. Ces chiffres sont essentiellement constitués par la vente de prestations de services. On peut dire que les logiciels libres sont déjà un standard de fait dans le domaine de l'infrastructure et qu'ils sont à l'aube d'une percée fulgurante dans le domaine applicatif. Ceci mérite amplement que l'on s'attarde un peu plus longtemps sur leur nature.

## Ouverture

Ce qui distingue un logiciel libre d'un logiciel propriétaire, c'est aussi un modèle de développement différent : un modèle plus ouvert et moins centralisé. C'est pourquoi on utilise aussi souvent en anglais le terme « Open Source » pour désigner les logiciels libres. Attention, c'est une expression parfois trompeuse, puisqu'il ne suffit pas que le code source d'un logiciel soit disponible pour qu'on puisse le qualifier de libre. Il faut encore que trois libertés fondamentales soient préservées pour chaque utilisateur : la possibilité de l'utiliser, de le modifier et de le distribuer.

Chaque logiciel libre, des plus petits, qui sont de simples utilitaires de quelques lignes de codes aux plus importants, tels que Linux ou Apache constitués de millions de lignes de codes, sont développés dans un laboratoire virtuel, un mélange de sites web, de listes de diffusion et d'autres moyens de communications que l'on trouve communément sur Internet.

Ce laboratoire virtuel, que les développeurs nomment projet, s'organise d'abord autour des initiateurs, puis si le logiciel est un succès, autour d'une association plus ou moins formelle. Le laboratoire le plus emblématique est ainsi le projet Linux ([www.kernel.org](http://www.kernel.org)<sup>4</sup>) qui développe le noyau du même nom coordonné par Linus Torvalds; le projet Mozilla qui développe le célèbre navigateur conçu à l'origine par Netscape. Consultez les sites web en « .org » sur Internet en y accolant devant le nom d'un logiciel libre et vous trouverez le plus souvent le projet associé ([www.zope.org](http://www.zope.org)<sup>5</sup>, [www.debian.org](http://www.debian.org)<sup>6</sup>, etc).

Il convient de préciser la notion que recouvre ici le terme de « projet ». Les relations que tissent les développeurs entre eux ont peu de choses en

commun avec celles qui s'imposent dans les « projets » logiciels très hiérarchisés que l'on rencontre au sein des entreprises et chez les éditeurs de logiciels propriétaires.

- Il y a d'abord peu de contacts physiques entre les participants qui communiquent presque exclusivement en utilisant les outils de l'Internet.
- Un projet logiciel libre est par essence le plus souvent international et la langue véhiculaire est majoritairement l'anglais même s'il existe des projets locaux et des équipes dédiées à chaque langue dans les grands projets.
- La réunion des développeurs est en fait une communauté d'intérêts : celui de voir progresser et s'améliorer l'architecture et les fonctionnalités d'un logiciel. Le moteur de la participation à un projet trouve racine en effet dans un besoin : le besoin d'une reconnaissance personnelle parfois, mais plus souvent le besoin dans une entreprise d'utiliser un logiciel qui puisse s'acquitter de telle ou telle tâche, ou encore plus pragmatique, le besoin d'améliorer un logiciel libre que l'on utilise déjà.
- La hiérarchie au sein du projet, lorsqu'elle existe, est surtout calquée sur la capacité de chacun à produire des lignes de codes répondant aux objectifs du projet. Ceux qui réclament un quelconque pouvoir dans l'orientation ou l'organisation du projet, doivent avoir déjà fait leurs preuves par une contribution de lignes de codes utilisables (les donneurs de leçon ont souvent droit à la réponse typique : « show me the code »).

Les acteurs qui se rencontrent dans un projet afin de créer un logiciel libre sont aussi bien des personnes physiques que des entreprises. Il y a aujourd'hui en effet de plus en plus de sociétés utilisatrices dont les intérêts convergent vers la réussite de tel ou tel logiciel libre et qui donc, naturellement, y participent. Cette contribution semble même parfois inconsciente puisqu'on a par exemple vu des employés de Microsoft, qui dans le cadre de leurs fonctions, développèrent un *patch* pour corriger le défaut d'un logiciel libre qu'ils utilisaient; la célèbre firme fabriquant Windows contribuant ainsi, certes modestement, à faire progresser « l'ennemi numéro 1 »<sup>7</sup>. Ne parlons pas de toutes les sociétés qui font basculer des millions de lignes de codes avec armes et bagages dans le camp du logiciel libre. Netscape fut l'une des premières au début de 1998, lorsqu'elle décida d'adopter le modèle du logiciel libre pour son navigateur. Mais aujourd'hui, bien d'autres lui ont emboîté le pas, citons à titre d'illustration la sortie d'OpenAdaptor, un logiciel libre d'EAI, par la Dresner Bank; OpenOffice, la suite bureautique par SUN; ou enfin Postfix, un serveur de messagerie par IBM.

## Qualité

La qualité des logiciels, c'est à dire entre autre la fiabilité, la scalabilité, est l'une des questions majeures à soulever si l'on désire comparer le logiciel libre au logiciel propriétaire. Il est admis que cette notion de qualité technique est d'une manière générale corrélée au faible nombre de bogues et à la capacité de les corriger efficacement.

Les défenseurs du logiciel libre déclarent que la combinaison des trois libertés (utilisation, modification et distribution) et du modèle de développement ouvert des logiciels libres exerce une pression constante en faveur d'une meilleure qualité. Mais comment est-il possible que les projets développant ces logiciels libres, structures informelles sur Internet, soient intrinsèquement plus compétitifs en terme de qualité que les programmes d'assurance qualité et autres processus formels de contrôle du développement des logiciels propriétaires chez les éditeurs?

Une première réponse à ce paradoxe apparent consiste à remarquer que les résultats obtenus par l'informatique libre parlent d'eux-mêmes. La fiabilité de certains logiciels libres est devenue célèbre : citons bien sûr Linux qui a été choisi aux États-Unis par la société SIAC, comme système d'exploitation de l'infrastructure du NYSE (New York Stock Exchange); ou encore simplement le site web le plus fréquenté au monde, Yahoo<sup>8</sup>, qui utilise depuis des années un autre système d'exploitation issu du monde des logiciels libres, FreeBSD. À l'opposé par exemple, les problèmes récurrents de stabilité agacent plus d'un administrateur de serveurs Windows 2000 ou Notes/Domino. La société Microsoft aurait elle-même reconnu que son logiciel Windows 2000 avait plus de soixante mille bogues<sup>9</sup>.

Mais au delà de ce constat, il faut bien comprendre que plusieurs facteurs concourent inexorablement à l'établissement d'une supériorité technique en terme qualitatif des logiciels libres.

- Les développeurs se réunissent un sein du projet par intérêt et dans une démarche volontaire : ne participent au projet que ceux qui sont motivés par la réussite du projet, et donc réciproquement, ceux qui participent au projet sont nécessairement motivés. Il est inutile d'insister sur le fait que des développeurs motivés sont bien plus efficaces pour prévenir ou découvrir des bogues.
- L'unique objectif d'un projet « logiciel libre » consiste dans la création du code source d'un logiciel : il n'y a pas d'interférence avec d'autres objectifs qui pourraient aller à l'encontre de la qualité de ce code; par exemple une opportunité de sortie (« release ») calquée sur une stratégie commerciale ou marketing comme c'est fréquent dans le monde du logiciel pro-



priétaire. Pour contrer un concurrent, rechercher des financements, profiter d'une saison favorable, on assiste souvent à des sorties prématurées de logiciels, parfois quasi inutilisables. Plus grave, certains éditeurs de logiciels propriétaires sont soupçonnés régulièrement d'éviter volontairement la correction de petits bogues pour rythmer et justifier des mises à jour payantes du logiciel.

- Au sein des projets importants qui conçoivent les logiciels libres, la concurrence pour gagner en prestige ou en influence est féroce. Cette compétition entre les développeurs s'exerce d'autant que la transparence sur le code source est totale : chacun peut scruter, analyser, décortiquer, commenter et critiquer le code des autres développeurs. Quoi de plus stimulant que de savoir que la qualité du code source sera surveillée et jugée par vos pairs et que de ce jugement dépendra votre influence et votre rôle dans le projet. La pression, inacceptable si elle n'était pas volontairement subie, est d'autant plus forte que seule une portion des développements proposés seront finalement autorisés à intégrer la version officielle du logiciel : les meilleurs. On peut qualifier ce processus de sélection Darwinienne, où le code source serait remplacé par l'ADN.
- Finalement la mobilisation des ressources pour écrire un logiciel libre est incompatible avec les finances d'un éditeur de logiciel propriétaire, aussi puissant soit-il. Il est en effet fréquent que plusieurs dizaines de développeurs aient réfléchi, codé, amélioré la même portion de code source : un procédé de contrôle qualité certes trop luxueux pour un éditeur propriétaire mais évidemment très efficace!

## Sécurité

Quelques détracteurs du logiciel libre avancent régulièrement l'idée selon laquelle le logiciel libre poserait des problèmes en terme de sécurité informatique. Cette affirmation mérite d'être étudiée précisément puisque la sécurité représente un enjeu majeur à l'heure où tous les systèmes ou presque sont interconnectés à Internet.

La divulgation massive des « entrailles du logiciel », le code source, aiderait les pirates dans leurs tentatives d'intrusion des systèmes. Cette idée qui peut sembler logique en première approche, n'est en fait qu'un mythe savamment entretenu contre le logiciel libre : elle ne tient pas compte des deux points fondamentaux suivants.

- Il faut rappeler que la sécurité en informatique est d'abord une question de protocoles et d'algorithmes mathématiques : on ne trouve pas dans

le code source des mots de passe « en clair » comme on trouverait une clef sous un paillason! La sécurité est donc d'autant mieux assurée que les protocoles et les algorithmes utilisés sont solides et bien sûr qu'ils ne comportent pas de failles, c'est à dire d'erreurs de programmation mettant en jeu la sécurité. Les avantages du logiciel libre qui tendent à une plus grande qualité du code, profitent donc aussi à la sécurité.

- S'il faut admettre que la publication du code source peut profiter à une personne mal intentionnée, elle permet aussi à de nombreux experts en sécurité indépendants du projet d'auditer le code source et donc de trouver et de corriger des failles de sécurité éventuelles. Le jeu perpétuel du glaive et du bouclier se retrouverait de nouveau à égalité si l'on ne prenait pas en compte qu'une grande part des audits ont lieu avant même la sortie officielle du logiciel, et donc avant que les failles deviennent une menace pour les utilisateurs.

À l'opposé, si la non publication des codes sources ne permet pas à des experts indépendants un audit sur la sécurité préalable à la sortie commerciale d'un logiciel propriétaire, elle n'empêche absolument pas un pirate d'exploiter une faille quelconque. À titre d'illustration, on citera le cas du logiciel Internet Explorer dont le code source n'est pas visible puisqu'il n'est pas un logiciel libre, mais qui défraie très régulièrement la chronique.

Sans porter d'accusation, n'est-on pas en droit de nous demander ce qui garantit vraiment que des logiciels propriétaires aussi répandus que le pare-feu FireWall-1 Check Point n'aient pas de *backdoor*? Quel éditeur est capable de l'affirmer sans fournir le code source du logiciel?

## Pérennité

L'avantage essentiel du logiciel libre n'est pas technique, il ne réside ni dans la meilleure qualité de son code, ni dans sa plus grande sécurité, mais dans sa pérennité inaliénable.

Une entreprise qui développe ou fait développer un logiciel spécifique pour son propre usage, a la possibilité de personnaliser complètement son outil pour répondre à ses besoins. Elle est aussi libre de choisir ses fournisseurs et de décider elle-même de l'opportunité ou de la cadence des évolutions. Mais cette liberté se paye littéralement au prix fort : elle doit supporter seule tous les coûts afférents au développement et à la maintenance de son logiciel.

À l'opposé, choisir un logiciel propriétaire chez un éditeur devrait permettre de partager les coûts de développement de l'outil informatique. La licence payée à l'éditeur équivaut en théorie à une portion plus ou moins partagée

du coût de développement du logiciel qui est ainsi mutualisé entre tous les acheteurs. Malheureusement la contrepartie de cette économie réside dans la dépendance trop forte qui se crée avec l'éditeur du logiciel propriétaire.

En effet, un logiciel n'est pas une voiture et on n'en change pas aussi facilement. Les données fabriquées par le logiciel ne sont pas facilement et immédiatement récupérables par un logiciel concurrent. La compréhension de son mode de fonctionnement ou de son interface nécessite des compétences ou une formation spécifique pour les utilisateurs. Enfin, l'enjeu d'interopérabilité avec toutes les autres composantes du système d'information de l'entreprise peut rendre coûteuse et délicate toute tentative de migration.

Si remplacer un logiciel propriétaire par son concurrent est déjà très complexe, le risque s'élève encore si l'on considère la vitesse d'évolution que connaît aujourd'hui le marché de l'informatique : obsolescence rapide des technologies et des modes, pression des éditeurs propriétaires pour ne fournir une assistance technique que pour les toutes dernières versions des logiciels, garanties contractuelles faibles ou presque inexistantes. Le client se trouve souvent en situation de payer des mises à jour qui corrigent en fait des vices cachés, ou pire avec un logiciel que l'éditeur ne maintient plus. L'abandon unilatéral de maintenance de la part d'un éditeur propriétaire est en effet de plus en plus fréquent et souvent prévu contractuellement!

Le piège se referme encore un peu plus de nos jours lorsque l'on constate que la nécessité croissante d'une interopérabilité globale - dans un contexte d'entreprise étendue, les clients, les fournisseurs et les partenaires d'une entreprise doivent communiquer avec les mêmes protocoles, les mêmes formats de données, etc - pousse chacun insidieusement vers des monopoles de fait. La pression de quelques grands éditeurs pour faire le choix « politiquement correct », sous peine de se retrouver isolé accentue ce phénomène. Les conséquences à terme pour les entreprises sont extrêmement coûteuses puisque cela entraîne moins de concurrence, et donc une hausse des tarifs de licence et une baisse de la qualité des logiciels propriétaires.

Le logiciel libre, parce qu'il favorise une factorisation du développement d'une part et qu'il est la garantie d'une émulation et d'une concurrence saine entre différents fournisseurs d'autre part, conjugue les avantages en terme d'indépendance et de personnalisation du logiciel « sur-mesure », avec la généricité et les économies du logiciel « sur l'étagère ».

Sa pérennité est donc inaliénable puisque qu'elle ne dépend plus de la survie ou des revirements d'un éditeur propriétaire, seul autorisé à lire et à modifier le code source du logiciel. L'entreprise utilisatrice aura toujours la liberté de mettre en compétition plusieurs fournisseurs sur la maintenance, la garantie ou l'évolution d'un logiciel libre qu'elle utilise.

Le logiciel libre mérite bien son nom : il libère l'entreprise de la relation déséquilibrée qu'entretiennent les éditeurs de logiciels propriétaires avec leurs clients.

## **Interopérabilité**

Beaucoup d'entreprises utilisent déjà les logiciels libres dans leur système d'information, au niveau des technologies de l'Internet notamment. Mais ce n'est sans aucun doute qu'un début puisque les avantages principaux du logiciel libre que nous avons constatés (qualité, sécurité, pérennité) sont à priori valables pour n'importe quel type de logiciel libre. Déjà, de nouveaux projets avec un modèle de développement ouvert se multiplient dans toutes les sphères de l'informatique. Les serveurs d'applications comme Zope et Jboss sont des bons exemples, mais c'est toute la sphère applicative qui voit fleurir les projets : OpenCRM, OpenERP, etc...

Certes, certains de ces nouveaux logiciels sont encore très jeunes et les avantages sur le long terme à adopter des logiciels libres ne sont pas encore suffisamment connus de tous. Mais il s'agit bien d'un changement fondamental qui se prépare dans la façon d'envisager le rapport de l'entreprise avec un objet immatériel comme le logiciel. Les lignes de codes, constituant la recette de cuisine du logiciel, sortent des greniers poussiéreux des éditeurs propriétaires pour s'afficher au yeux de tous, prêtes à être modifiées par les meilleurs développeurs du monde entier. Le code source est en train de subir le même sort que les protocoles et autres normes qui sont d'ores et déjà créés, modifiés et partagés par tous les acteurs du marché. Pour le bien de tous, les mêmes algorithmes ne seront pas sans cesse réinventés. C'est donc la promesse pour les utilisateurs qu'ils pourront bénéficier d'une véritable interopérabilité dans le cadre d'une concurrence préservée.

## **Conclusion**

Les logiciels libres vont continuer à progresser plus vite que leurs équivalents propriétaires aussi sûrement qu'une boule de neige qui dévale une pente n'arrête pas de grossir. Effectivement, plutôt que de payer cher un logiciel chez un éditeur propriétaire, avec les risques de dépendance que cela entraîne, il y aura toujours une entreprise pour initier le développement de fonctionnalités additionnelles qui manquent dans les logiciels libres existants.

# Glossaire

## **bogue**

Traduction française de « bug ». Un bogue est une erreur dans la programmation d'un logiciel qui entraîne des dysfonctionnements.

## **patch**

Morceau de code source devant être ajouté à un logiciel pour corriger un bogue ou ajouter une fonctionnalité. On utilise parfois la traduction française « rustine ».

## **backdoor**

Porte d'entrée dérobée et non-documentée dans un système informatique. Ce moyen d'accès peut avoir été volontairement ajouté à un logiciel ou peut résulter d'un bogue. Cette fonctionnalité cachée est potentiellement très dangereuse puisque des pirates finiront pas la découvrir.

## Notes

1. <http://www.alcove.com/>
2. Étude du mois de septembre 2001 de Security Space  
([http://www.securityspace.com/s\\_survey/data/200108/index.html](http://www.securityspace.com/s_survey/data/200108/index.html))
3. Voir par exemple l'article de Zdnet « Is Microsoft secretly using open source? »  
(<http://www.zdnet.com/zdnn/stories/news/0,4586,2776342,00.html?>)
4. <http://www.kernel.org/>
5. <http://www.zope.org/>
6. <http://www.debian.org/>
7. Techweb « Ballmer: Linux Is Top Threat To Windows »  
(<http://slashdot.org/articles/01/01/10/1947202.shtml>)
8. <http://www.yahoo.com/>
9. Zdnet News : « Bugfest! Win2000 has 63,000 'defects' »  
(<http://www.zdnet.com/zdnn/stories/news/0,4586,2436920,00.html?>)