

## Le monde des logiciels libres

Un aperçu du libre, de ses points faibles, de ses points forts et de ses enjeux

Adrien Destugues

Forum ENSSAT 2010

"This resembles the relationship between Newtonian and Einsteinian physics - the older system is still valid at low energies, but if you push mass and velocity high enough you get surprises like nuclear explosions or Linux."

# Plan

- 1 Présentation
- 2 Les logiciels libres
- 3 Le libre et l'économie
- 4 La communauté du logiciel libre
- 5 Conclusion

# Plan

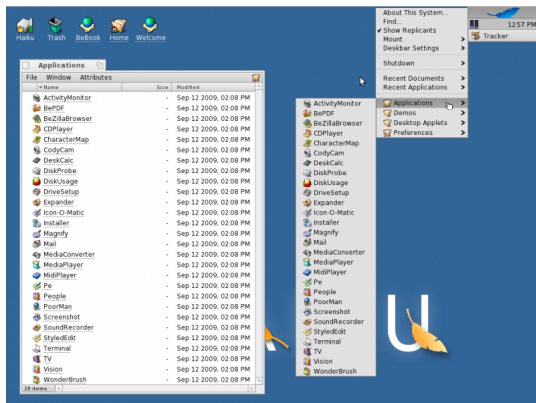
- 1 **Présentation**
- 2 Les logiciels libres
- 3 Le libre et l'économie
- 4 La communauté du logiciel libre
- 5 Conclusion



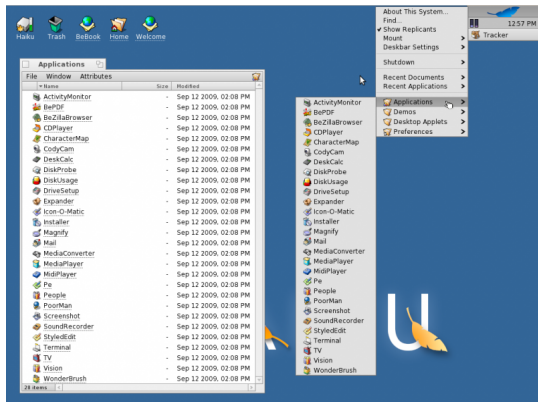
- Adrien Destugues, élève ingénieur ENSSAT EII2
- Membre du projet Haiku depuis mai 2009
- Contributeur de plusieurs autres projets libres



- Un système d'exploitation (comme Windows ou Linux)
- Inspiré et compatible avec BeOS
- Destiné à une utilisation sur un ordinateur personnel
- En développement depuis 2001, première version alpha en septembre 2009

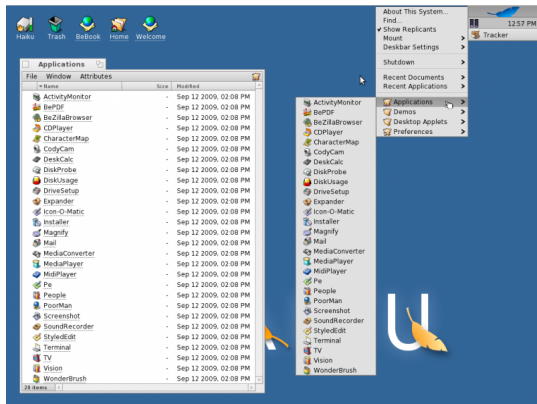


- Un système d'exploitation (comme Windows ou Linux)
- Inspiré et compatible avec BeOS
- Destiné à une utilisation sur un ordinateur personnel
- En développement depuis 2001, première version alpha en septembre 2009

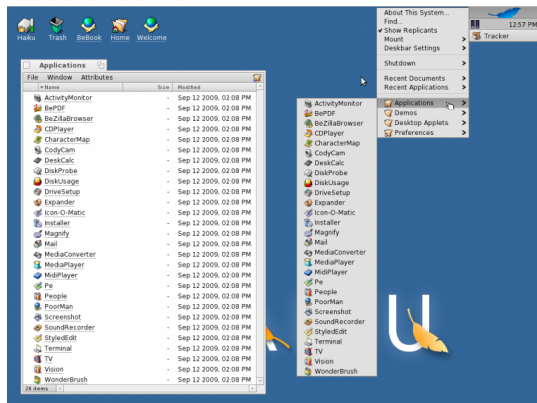




- Un système d'exploitation (comme Windows ou Linux)
- Inspiré et compatible avec BeOS
- Destiné à une utilisation sur un ordinateur personnel
- En développement depuis 2001, première version alpha en septembre 2009



- Un système d'exploitation (comme Windows ou Linux)
- Inspiré et compatible avec BeOS
- Destiné à une utilisation sur un ordinateur personnel
- En développement depuis 2001, première version alpha en septembre 2009



# Plan

- 1 Présentation
- 2 Les logiciels libres**
- 3 Le libre et l'économie
- 4 La communauté du logiciel libre
- 5 Conclusion

- **Années 60 et 70 : le matériel est vendu, le logiciel est libre en l'absence de législation**
- 1976 : naissance de la micro-informatique et début de la séparation entre logiciel et matériel
- 1981 : la licence MS-DOS
- 1983 : le projet GNU
- 1991 : Linux

- **Années 60 et 70 : le matériel est vendu, le logiciel est libre en l'absence de législation**
- **1976 : naissance de la micro-informatique et début de la séparation entre logiciel et matériel**
- 1981 : la licence MS-DOS
- 1983 : le projet GNU
- 1991 : Linux

- Années 60 et 70 : le matériel est vendu, le logiciel est libre en l'absence de législation
- 1976 : naissance de la micro-informatique et début de la séparation entre logiciel et matériel
- 1981 : la licence MS-DOS
- 1983 : le projet GNU
- 1991 : Linux

- Années 60 et 70 : le matériel est vendu, le logiciel est libre en l'absence de législation
- 1976 : naissance de la micro-informatique et début de la séparation entre logiciel et matériel
- 1981 : la licence MS-DOS
- 1983 : le projet GNU
- 1991 : Linux





## Liberté 0

Utiliser le logiciel

Contre exemples : Google Documents, Windows XP

## Liberté 1

Étudier le logiciel

Accès au code source

## Liberté 2

Distribuer et partager le logiciel

## Liberté 3

Modifier le logiciel et distribuer les versions modifiées

## Liberté 0

Utiliser le logiciel

Contre exemples : Google Documents, Windows XP

## Liberté 1

Étudier le logiciel

Accès au code source

## Liberté 2

Distribuer et partager le logiciel

## Liberté 3

Modifier le logiciel et distribuer les versions modifiées

## Liberté 0

Utiliser le logiciel

Contre exemples : Google Documents, Windows XP

## Liberté 1

Étudier le logiciel

Accès au code source

## Liberté 2

Distribuer et partager le logiciel

## Liberté 3

Modifier le logiciel et distribuer les versions modifiées

## Liberté 0

Utiliser le logiciel

Contre exemples : Google Documents, Windows XP

## Liberté 1

Étudier le logiciel

Accès au code source

## Liberté 2

Distribuer et partager le logiciel

## Liberté 3

Modifier le logiciel et distribuer les versions modifiées

# La licence GPL

- Créée par la Free Software Foundation
- Environ 70% des logiciels libres
- Liberté du programme avant celle du programmeur :  
lorsqu'on modifie un programme, on est obligé de publier  
les changements avec la même licence.
- Texte assez compliqué...

# La licence GPL

- Créée par la Free Software Foundation
- Environ 70% des logiciels libres
- Liberté du programme avant celle du programmeur : lorsqu'on modifie un programme, on est obligé de publier les changements avec la même licence.
- Texte assez compliqué...

# La licence GPL

- Créée par la Free Software Foundation
- Environ 70% des logiciels libres
- Liberté du programme avant celle du programmeur :  
lorsqu'on modifie un programme, on est obligé de publier  
les changements avec la même licence.
- Texte assez compliqué...

# La licence GPL

- Créée par la Free Software Foundation
- Environ 70% des logiciels libres
- Liberté du programme avant celle du programmeur :  
lorsqu'on modifie un programme, on est obligé de publier  
les changements avec la même licence.
- Texte assez compliqué...



# La licence BSD/MIT

- Créée par l'université de Californie
- Liberté du programmeur avant celle du programme : tout est autorisé
- Il est possible de mélanger du code BSD avec du code non libre ou sous GPL
- Texte simple : seulement 5 lignes

# La licence BSD/MIT

- Créée par l'université de Californie
- Liberté du programmeur avant celle du programme : tout est autorisé
- Il est possible de mélanger du code BSD avec du code non libre ou sous GPL
- Texte simple : seulement 5 lignes

# La licence BSD/MIT

- Créée par l'université de Californie
- Liberté du programmeur avant celle du programme : tout est autorisé
- Il est possible de mélanger du code BSD avec du code non libre ou sous GPL
- Texte simple : seulement 5 lignes

# La licence BSD/MIT

- Créée par l'université de Californie
- Liberté du programmeur avant celle du programme : tout est autorisé
- Il est possible de mélanger du code BSD avec du code non libre ou sous GPL
- Texte simple : seulement 5 lignes

# Les autres licences

- Le domaine public : renoncement complet aux droits d'auteur (variable selon les pays)
- La licence LGPL : GPL adaptée aux bibliothèques de code
- Affero : une licence adaptée à l'utilisation de logiciels en ligne (sites internet, ...)

# Les autres licences

- Le domaine public : renoncement complet aux droits d'auteur (variable selon les pays)
- La licence LGPL : GPL adaptée aux bibliothèques de code
- Affero : une licence adaptée à l'utilisation de logiciels en ligne (sites internet, ...)

# Les autres licences

- Le domaine public : renoncement complet aux droits d'auteur (variable selon les pays)
- La licence LGPL : GPL adaptée aux bibliothèques de code
- Affero : une licence adaptée à l'utilisation de logiciels en ligne (sites internet, ...)

Un logiciel libre est toujours open source,  
Un logiciel open source n'est pas toujours libre.  
L'open source ne garantit que la liberté n°2 : étudier le programme.



# Plan

- 1 Présentation
- 2 Les logiciels libres
- 3 Le libre et l'économie**
- 4 La communauté du logiciel libre
- 5 Conclusion

- Redéfinir le modèle économique du logiciel
- Copier et distribuer un logiciel ne coûte rien, le concevoir demande du travail
- Acheter du code (ou du temps de travail) à une entreprise ou à un programmeur, puis en faire ce qu'on veut

- Redéfinir le modèle économique du logiciel
- Copier et distribuer un logiciel ne coûte rien, le concevoir demande du travail
- Acheter du code (ou du temps de travail) à une entreprise ou à un programmeur, puis en faire ce qu'on veut

- Redéfinir le modèle économique du logiciel
- Copier et distribuer un logiciel ne coûte rien, le concevoir demande du travail
- Acheter du code (ou du temps de travail) à une entreprise ou à un programmeur, puis en faire ce qu'on veut

- Vendre un logiciel libre est difficile : on autorise l'acheteur à le distribuer gratuitement.
- Vendre des services reste possible : support technique, développement d'applications spécifiques. . .
- Les licences de type BSD permettent une évolution progressive

- Vendre un logiciel libre est difficile : on autorise l'acheteur à le distribuer gratuitement.
- Vendre des services reste possible : support technique, développement d'applications spécifiques. . .
- Les licences de type BSD permettent une évolution progressive

- Vendre un logiciel libre est difficile : on autorise l'acheteur à le distribuer gratuitement.
- Vendre des services reste possible : support technique, développement d'applications spécifiques. . .
- Les licences de type BSD permettent une évolution progressive





# Travailler pour le libre

- Rejoindre une SSLL,
- Participer à des programmes comme le Google Summer of Code,
- Travailler directement pour un projet (FSF, Firefox, Haiku, ...)
- Donations et "bounties"

# Travailler pour le libre

- Rejoindre une SSSL,
- Participer à des programmes comme le Google Summer of Code,
- Travailler directement pour un projet (FSF, Firefox, Haiku, ...)
- Donations et "bounties"

# Travailler pour le libre

- Rejoindre une SSLL,
- Participer à des programmes comme le Google Summer of Code,
- Travailler directement pour un projet (FSF, Firefox, Haiku, ...)
- Donations et "bounties"

## Certains projets libres ne sont pas des logiciels

- Wikipedia
- Open Cola, Free Beer
- Touchbook, matériel libre

## Certains projets libres ne sont pas des logiciels

- Wikipedia
- Open Cola, Free Beer
- Touchbook, matériel libre

## Certains projets libres ne sont pas des logiciels

- Wikipedia
- Open Cola, Free Beer
- Touchbook, matériel libre

- Un logiciel propriétaire peut devenir libre : Warzone 2100, Firefox, Blender, ...
- Un logiciel libre ne peut pas devenir propriétaire : Wine
- Un logiciel libre peut copier et dépasser un logiciel propriétaire : ReactOS, Haiku
- Un logiciel libre ne peut pas disparaître, en particulier grâce à Internet

- Un logiciel propriétaire peut devenir libre : Warzone 2100, Firefox, Blender, ...
- Un logiciel libre ne peut pas devenir propriétaire : Wine
- Un logiciel libre peut copier et dépasser un logiciel propriétaire : ReactOS, Haiku
- Un logiciel libre ne peut pas disparaître, en particulier grâce à Internet



- Un logiciel propriétaire peut devenir libre : Warzone 2100, Firefox, Blender, ...
- Un logiciel libre ne peut pas devenir propriétaire : Wine
- Un logiciel libre peut copier et dépasser un logiciel propriétaire : ReactOS, Haiku
- Un logiciel libre ne peut pas disparaître, en particulier grâce à Internet

- Un logiciel propriétaire peut devenir libre : Warzone 2100, Firefox, Blender, ...
- Un logiciel libre ne peut pas devenir propriétaire : Wine
- Un logiciel libre peut copier et dépasser un logiciel propriétaire : ReactOS, Haiku
- Un logiciel libre ne peut pas disparaître, en particulier grâce à Internet

# Plan

- 1 Présentation
- 2 Les logiciels libres
- 3 Le libre et l'économie
- 4 La communauté du logiciel libre**
- 5 Conclusion

- Un cœur assez réduit, de 3 à 40 développeurs selon le projet
- Beaucoup de contributeurs externes
- Une communauté d'utilisateurs actifs
- Communication essentiellement par Internet dans la plupart des cas

- Un cœur assez réduit, de 3 à 40 développeurs selon le projet
- Beaucoup de contributeurs externes
- Une communauté d'utilisateurs actifs
- Communication essentiellement par Internet dans la plupart des cas

- Un cœur assez réduit, de 3 à 40 développeurs selon le projet
- Beaucoup de contributeurs externes
- Une communauté d'utilisateurs actifs
- Communication essentiellement par Internet dans la plupart des cas

- Un cœur assez réduit, de 3 à 40 développeurs selon le projet
- Beaucoup de contributeurs externes
- Une communauté d'utilisateurs actifs
- Communication essentiellement par Internet dans la plupart des cas







- Défi personnel, quête de reconnaissance
- Besoin particulier d'une application
- Publication de "vieux" code pour assurer la survie d'un logiciel
- Partage du savoir (laboratoires de recherche publics)
- Rarement l'argent . . .

- Défi personnel, quête de reconnaissance
- Besoin particulier d'une application
- Publication de "vieux" code pour assurer la survie d'un logiciel
- Partage du savoir (laboratoires de recherche publics)
- Rarement l'argent . . .

- Défi personnel, quête de reconnaissance
- Besoin particulier d'une application
- Publication de "vieux" code pour assurer la survie d'un logiciel
- Partage du savoir (laboratoires de recherche publics)
- Rarement l'argent . . .

- En tant qu'utilisateur : envoyer un rapport de bug ou faire une suggestion
- En tant que développeur : envoyer un "patch"
- Pour entrer dans le cœur du projet : co-option par les membres du projet
- Ou bien créer son propre projet. . .

- En tant qu'utilisateur : envoyer un rapport de bug ou faire une suggestion
- En tant que développeur : envoyer un "patch"
- Pour entrer dans le cœur du projet : co-option par les membres du projet
- Ou bien créer son propre projet. . .

- En tant qu'utilisateur : envoyer un rapport de bug ou faire une suggestion
- En tant que développeur : envoyer un "patch"
- Pour entrer dans le cœur du projet : co-option par les membres du projet
- Ou bien créer son propre projet. . .

- En tant qu'utilisateur : envoyer un rapport de bug ou faire une suggestion
- En tant que développeur : envoyer un "patch"
- Pour entrer dans le cœur du projet : co-option par les membres du projet
- Ou bien créer son propre projet. . .





- Fort couplage entre les projets : noyau linux + outils gnu + Xorg + ... = un système utilisable
- Une séparation assez nette entre logiciels "GPL" et "BSD", mais les deux sont indispensables
- Les "fork" : copie du code source d'un logiciel pour démarrer un nouveau projet. Duplication d'effort inutile, ou source de diversité ?



- Les cycles de publication : date ou liste de fonctionnalités, mais rarement les deux
- La maintenance : branches stable et de développement
- Le contrôle de qualité : évaluation par les pairs (plus ils sont nombreux, plus il y a de chance de trouver la faille)
- La distribution : faite par les utilisateurs

- Les cycles de publication : date ou liste de fonctionnalités, mais rarement les deux
- La maintenance : branches stable et de développement
- Le contrôle de qualité : évaluation par les pairs (plus ils sont nombreux, plus il y a de chance de trouver la faille)
- La distribution : faite par les utilisateurs

- Les cycles de publication : date ou liste de fonctionnalités, mais rarement les deux
- La maintenance : branches stable et de développement
- Le contrôle de qualité : évaluation par les pairs (plus ils sont nombreux, plus il y a de chance de trouver la faille)
- La distribution : faite par les utilisateurs

- Les cycles de publication : date ou liste de fonctionnalités, mais rarement les deux
- La maintenance : branches stable et de développement
- Le contrôle de qualité : évaluation par les pairs (plus ils sont nombreux, plus il y a de chance de trouver la faille)
- La distribution : faite par les utilisateurs

- La communication interne : centralisée (listes de diffusions), en temps décalé (mail plutôt que téléphone), publique
- L'innovation : en général elle vient d'une seule personne. Plus il y a de gens, plus il y a d'idées. La réalisation peut être collective
- La vision à long terme : s'il y a désaccord, on peut "forker" le projet



- La communication interne : centralisée (listes de diffusions), en temps décalé (mail plutôt que téléphone), publique
- L'innovation : en général elle vient d'une seule personne. Plus il y a de gens, plus il y a d'idées. La réalisation peut être collective
- La vision à long terme : s'il y a désaccord, on peut "forker" le projet

- La communication interne : centralisée (listes de diffusions), en temps décalé (mail plutôt que téléphone), publique
- L'innovation : en général elle vient d'une seule personne. Plus il y a de gens, plus il y a d'idées. La réalisation peut être collective
- La vision à long terme : s'il y a désaccord, on peut "forker" le projet

- Difficilement applicable au contenu artistique (livres, jeux vidéos, ...)
- Complication de la maintenance (si le client modifie le logiciel par rapport à la version connue)
- Éventuelle prolifération des versions et des alternatives pour un même problème

- Difficilement applicable au contenu artistique (livres, jeux vidéos, ...)
- Complication de la maintenance (si le client modifie le logiciel par rapport à la version connue)
- Éventuelle prolifération des versions et des alternatives pour un même problème

- Difficilement applicable au contenu artistique (livres, jeux vidéos, ...)
- Complication de la maintenance (si le client modifie le logiciel par rapport à la version connue)
- Éventuelle prolifération des versions et des alternatives pour un même problème

# Plan

- 1 Présentation
- 2 Les logiciels libres
- 3 Le libre et l'économie
- 4 La communauté du logiciel libre
- 5 Conclusion**

- 1 "Le libre, c'est des gens qui veulent la fin de Microsoft"
- 2 "Le libre, c'est des informaticiens barbus qui font des logiciels bizarres et le plus souvent inutiles"
- 3 "Le libre, c'est un truc de pirates"
- 4 "Le libre, c'est un truc de communistes"
- 5 "Un logiciel libre est un logiciel gratuit"
- 6 "Un logiciel libre est forcément moins bien que son équivalent commercial"

- 1 "Le libre, c'est des gens qui veulent la fin de Microsoft"
- 2 "Le libre, c'est des informaticiens barbus qui font des logiciels bizarres et le plus souvent inutiles"
- 3 "Le libre, c'est un truc de pirates"
- 4 "Le libre, c'est un truc de communistes"
- 5 "Un logiciel libre est un logiciel gratuit"
- 6 "Un logiciel libre est forcément moins bien que son équivalent commercial"



- 1 "Le libre, c'est des gens qui veulent la fin de Microsoft"
- 2 "Le libre, c'est des informaticiens barbus qui font des logiciels bizarres et le plus souvent inutiles"
- 3 "Le libre, c'est un truc de pirates"
- 4 "Le libre, c'est un truc de communistes"
- 5 "Un logiciel libre est un logiciel gratuit"
- 6 "Un logiciel libre est forcément moins bien que son équivalent commercial"

- 1 "Le libre, c'est des gens qui veulent la fin de Microsoft"
- 2 "Le libre, c'est des informaticiens barbus qui font des logiciels bizarres et le plus souvent inutiles"
- 3 "Le libre, c'est un truc de pirates"
- 4 "Le libre, c'est un truc de communistes"
- 5 "Un logiciel libre est un logiciel gratuit"
- 6 "Un logiciel libre est forcément moins bien que son équivalent commercial"

- 1 "Le libre, c'est des gens qui veulent la fin de Microsoft"
- 2 "Le libre, c'est des informaticiens barbus qui font des logiciels bizarres et le plus souvent inutiles"
- 3 "Le libre, c'est un truc de pirates"
- 4 "Le libre, c'est un truc de communistes"
- 5 "Un logiciel libre est un logiciel gratuit"
- 6 "Un logiciel libre est forcément moins bien que son équivalent commercial"

- 1 "Le libre, c'est des gens qui veulent la fin de Microsoft"
- 2 "Le libre, c'est des informaticiens barbus qui font des logiciels bizarres et le plus souvent inutiles"
- 3 "Le libre, c'est un truc de pirates"
- 4 "Le libre, c'est un truc de communistes"
- 5 "Un logiciel libre est un logiciel gratuit"
- 6 "Un logiciel libre est forcément moins bien que son équivalent commercial"

- 1 Un nouveau modèle économique
- 2 Un changement irréversible
- 3 Une intégration progressive
- 4 Un développement lié à Internet

- 1 Un nouveau modèle économique
- 2 Un changement irréversible
- 3 Une intégration progressive
- 4 Un développement lié à Internet

- 1 Un nouveau modèle économique
- 2 Un changement irréversible
- 3 Une intégration progressive
- 4 Un développement lié à Internet

- 1 Un nouveau modèle économique
- 2 Un changement irréversible
- 3 Une intégration progressive
- 4 Un développement lié à Internet