



# Dissemin : un nouvel outil pour l'accès ouvert (voie verte)

Antonin Delpéuch  
étudiant à l'École normale supérieure

<http://dissem.in/>

# Équipe



pintoch



bThom



Evarin



Marie



a3nm



p4bl0



LuGas

equipe@dissem.in  
@disseminOA

# Plan

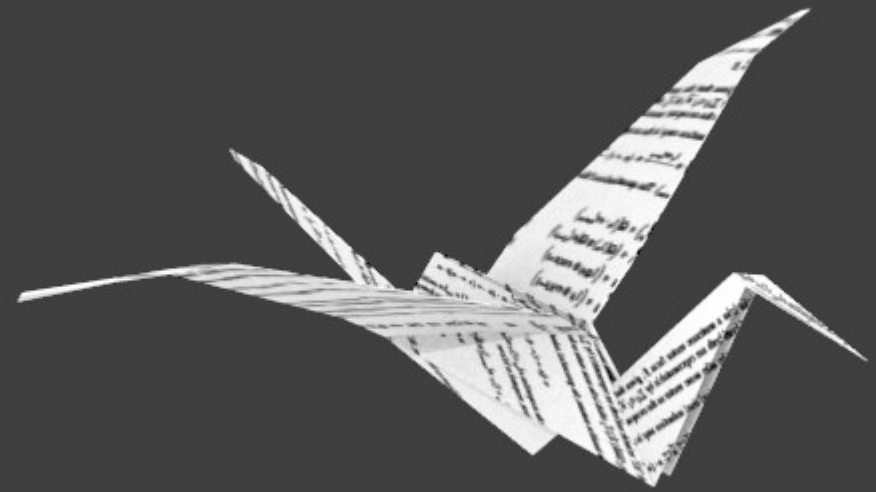
1. Le problème de l'accès fermé, vu par les chercheurs
2. Les archives ouvertes : une solution ?
3. Démo de notre plateforme facilitant les dépôts



equipe@dissem.in - <http://dissem.in/> - @disseminOA

# Plan

1. Le problème de l'accès fermé, vu par les chercheurs
2. Les archives ouvertes : une solution ?
3. Démo de notre plateforme facilitant les dépôts



equipe@dissem.in - <http://dissem.in/> - @disseminOA

# Trouver des publications ? Facile !

 **Preview**

 **PDF**

**Access options**

## Download:

- PDF
- PostScript
- Other formats

PMCID: PMC2515340

[Article](#) [PubReader](#) [PDF-785K](#) [Citation](#)

» [Download PDF](#) (191KB)



[Download full text in PDF](#)

↓ **DOWNLOAD**



[Get PDF](#) (215K)

[\[PDF\]](#) à partir de [cwi.nl](#)

[Article](#) | [PubReader](#) | [ePub \(beta\)](#) | [PDF \(469K\)](#) | [Citation](#)

*Proc. ICALP*, p. 56-68, Kyoto, Japon, juillet 2015. ([pdf](#) | [slides](#) | [bib](#))

[Download full-text](#)



[Download Book](#) (PDF, 4939 KB)

Sums of digits, overlaps, and  
*Discrete Math. & Theoret.*  
([Prétirage/Preprint](#))



[Download Chapter](#) (244 KB)

Sign up for a free account to **access more full-texts.**

[Join for free](#)

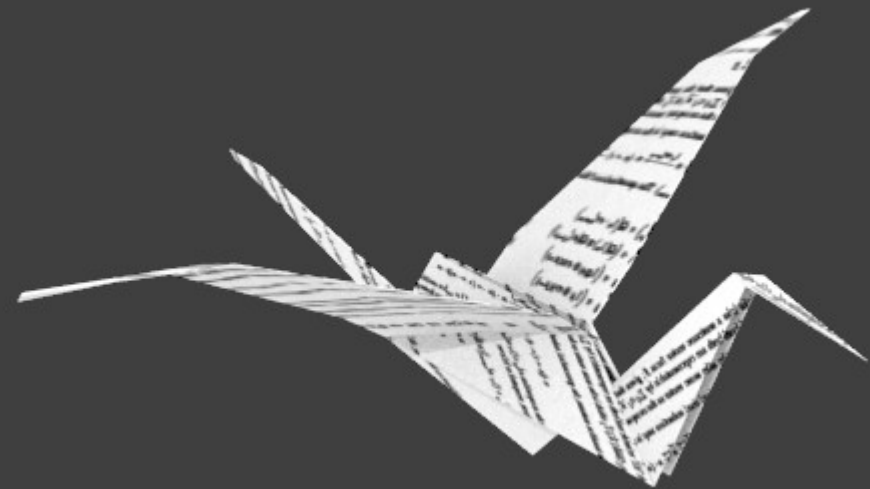
[Article](#) |  [PDF \(1251 K\)](#) | [View Record in Scopus](#) | [Citing articles \(5\)](#)

# Pourquoi achète-t-on encore des abonnements si coûteux ?

- Culture de l'auto-archivage peu répandue dans certains domaines ;
- Difficulté d'estimer le besoin réel de ces abonnements : les sources alternatives sont éparpillées et peu structurées ;
- Abonnements groupés qui réduisent la marge de manœuvre des négociateurs.

# Plan

1. Le problème de l'accès fermé, vu par les chercheurs
2. Les archives ouvertes : une solution ?
3. Démo de notre plateforme facilitant les dépôts



equipe@dissem.in - <http://dissem.in/> - @disseminOA

# La solution : les archives ouvertes

- Stockage pérenne des articles ;
- Formatage standardisé des métadonnées ;
- Indexation sans bavures.



# Pourquoi si peu d'articles dans les dépôts ?

- **Longs formulaires** à remplir, délais de modération ;
- Doutes des chercheurs sur leurs **droits** à déposer ;
- Méfiance vis-à-vis des dépôts utilisés comme **vitrines bibliométriques** des institutions ;
- Molesse ou inadéquation des **incitations institutionnelles** ;
- Attrait des **pages personnelles** ou **réseaux sociaux académiques**.

# Pourquoi si peu d'articles dans les dépôts ?

## Problèmes ?

- **Longs formulaires ;**
- **Droit** de déposer ;
- Méfiance vis-à-vis de la **bibliométrie** ;
- Inadéquation des **incitations**.

## Solutions !

- **Métadonnées préremplies ;**
- **Analyse claire** des politiques ;
- **Indépendance** de la plateforme, omission des affiliations ;
- Incitation **simple** et visuelle : augmenter la proportion d'articles disponibles.

# Plan

1. Le problème de l'accès fermé, vu par les chercheurs
2. Les archives ouvertes : une solution ?
3. Démo de notre plateforme facilitant les dépôts



equipe@dissem.in - <http://dissem.in/> - @disseminOA

## Bienvenue sur dissemin

Dissemin est un service gratuit pour aider les chercheurs à vérifier que leurs publications sont librement accessibles pour leurs lecteurs. Notre service identifie les papiers qui ne peuvent être obtenus qu'avec des souscriptions payantes, et vous permet de les mettre en ligne en quelques clics sur [Zenodo](#), un dépôt innovant soutenu par l'UE.



Vous hésitez toujours ? Lisez ce qui suit ou jetez un œil à la [FAQ](#).

Analyser mes publications

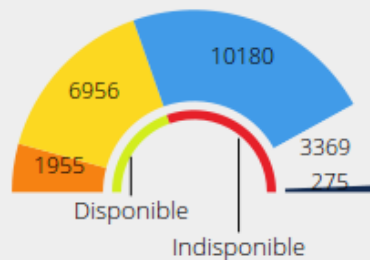
ou

Chercher un collègue

### Libre accès (voie verte)

Les chercheurs ont souvent le droit de mettre leurs articles en accès libre sur le Web, pour compléter la version payante proposée par les éditeurs traditionnels. Pourtant, tous ne le font pas.

À cause de cela, les bibliothèques doivent payer à prix d'or des abonnements électroniques aux journaux de recherche, ce qui grève leurs finances et limite leur offre.



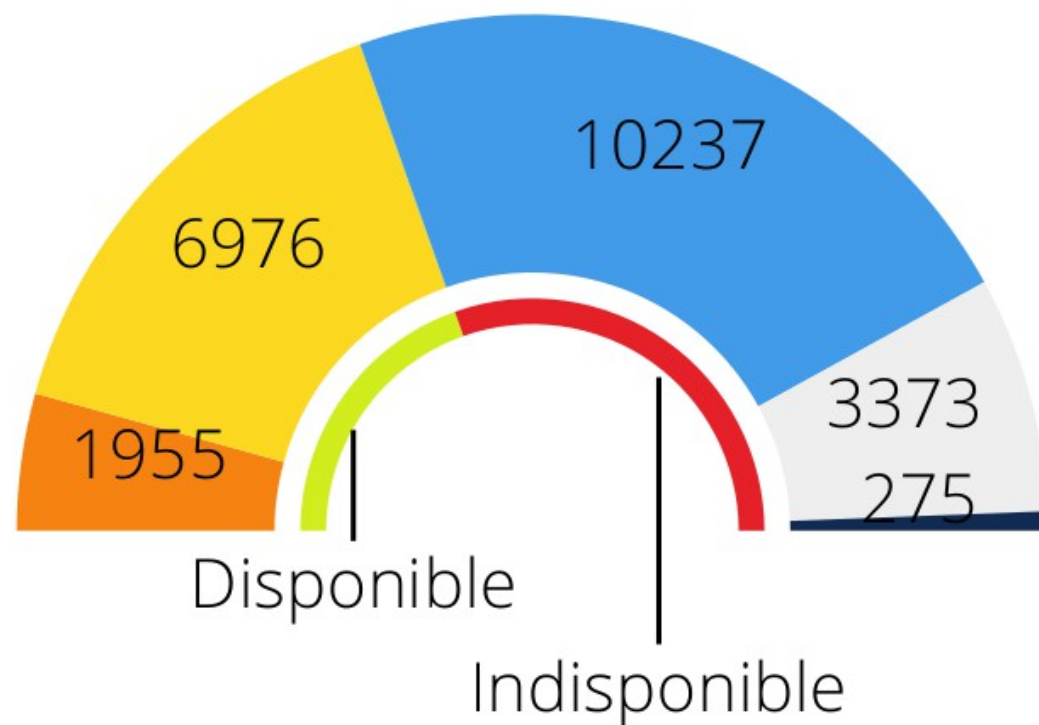
Accessible à partir de l'éditeur. (1955)

Accessible à partir de l'auteur (6956)

### Dépôts ouverts

C'est déjà bien de mettre vos articles en ligne sur votre page Web, mais ce n'est pas suffisant ! De telles copies sont moins pérennes et plus difficiles à trouver que celles qui sont déposées dans des dépôts bien indexés.

Dissemin cherche des copies de vos articles dans une grande collection de dépôts ouverts en ligne, et vous indique ceux qu'il n'a pas pu trouver de cette façon.



- Accessible à partir de l'éditeur (1955)
- Accessible à partir de l'auteur (6976)
- Pourrait être partagé par les auteurs (10237)
- Politique inconnue ou complexe (3373)
- Partage interdit par l'éditeur (275)

# Bienvenue sur dissemin

Dissemin est un service gratuit pour aider les chercheurs à vérifier que leurs publications sont librement accessibles pour leurs lecteurs. Notre service identifie les papiers qui ne peuvent être obtenus qu'avec des souscriptions payantes, et vous permet de les mettre en ligne en quelques clics sur [Zenodo](#), un dépôt innovant soutenu par l'UE.



Vous hésitez toujours ? Lisez ce qui suit ou jetez un œil à la [FAQ](#).

Analyser mes publications

ou

Chercher un collègue

*par nom*

Peter

Sewell

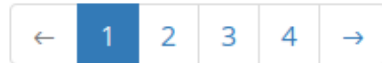
*ou par ORCID*

identifiant (par exemple, 0000-0001-2345-6789)

Recherche

## Papiers écrits par Peter Sewell

Ce chercheur n'est pas associé à un profil ORCID. Les publications ci-dessous correspondent au nom mais peuvent ne pas être pertinentes.



David Kaloper-Meršinjak, Hannes Mehnert, Anil Madhavapeddy, **Peter Sewell**

2015

Supplementary material for the publication "Not-quite-so-broken TLS: lessons in re-engineering a security protocol specification and implementation"

[Télécharger](#) | Zenodo.



Mark Batty, Kayvan Memarian, Kyndylan Nienhuis, Jean Pichon-Pharabod, **Peter Sewell**

The Problem of Programming Language Concurrency Semantics

[Mettre en ligne](#) | Springer Verlag, Lecture Notes in Computer Science, 2015.



Dominic P. Mulligan, Kathryn E. Gray, Scott Owens, Tom Ridge, **Peter Sewell**

2014

Lem

[Télécharger](#) | Proceedings of the 19th ACM SIGPLAN international conference on Functional programming - ICFP '14, 2014.



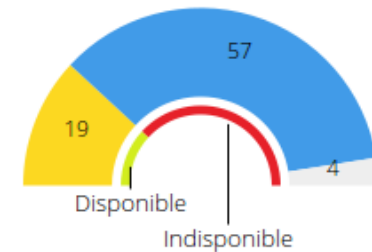
**Peter Sewell**

POPL 2014 program chair's report

[Mettre en ligne](#) | ACM SIGPLAN Notices, 4(49), 2014.

### Personne

#### Peter Sewell



- Accessible à partir de l'éditeur. (0)
- Accessible à partir de l'auteur (19)
- Pourrait être partagé par les auteurs (57)
- Politique inconnue ou complexe (4)
- Partage interdit par l'éditeur (0)

### Affiner la recherche

Par type de document:

- Article dans une revue
- Communication dans une conférence
- Chapitre d'ouvrage
- Ouvrage
- Numéro d'une revue
- Actes de conférence
- Article d'encyclopédie
- Poster
- Rapport
- Thèse
- Banque de données
- Préprint
- Autre publication

# Mathematizing C++ concurrency

Article dans une revue de Mark Batty, Scott Owens, Susmit Sarkar, Peter Sewell, Tjark Weber



**Texte intégral:** Indisponible

**Maison d'édition:** Association for Computing Machinery (ACM)



Prépublication : dépôt autorisé.

Mettre en ligne



Postdocument : dépôt autorisé.

Mettre en ligne



Version publiée : dépôt interdit.

[Policy details \(opens in a new window\).](#)

Data provided by SHERPA/ROMEO

## Publié dans

Proceedings of the 38th annual ACM SIGPLAN-SIGACT symposium on Principles of programming languages - POPL '11, 2011

DOI: 10.1145/1926385.1926394

ACM SIGPLAN Notices, **1**(46), 2011

DOI: 10.1145/1925844.1926394

## Liens

[\[kar.kent.ac.uk\]](http://kar.kent.ac.uk)

[Association for Computing Machinery \(ACM\)](#) via DOI

[Association for Computing Machinery \(ACM\)](#) via DOI

## Outils

[Chercher dans Google Scholar](#)

[Chercher dans CORE](#)

## Résumé

Shared-memory concurrency in C and C++ is pervasive in systems programming, but has long been poorly defined. This motivated an ongoing shared effort by the standards committees to specify concurrent behaviour in the next versions of both languages. They aim to provide strong guarantees for race-free programs, together with new (but subtle) relaxed-memory atomic primitives for high-performance concurrent code. However, the current draft standards, while the result of careful deliberation, are not yet clear and rigorous definitions, and harbour substantial problems in their details. In this paper we establish a mathematical (yet readable) semantics for C++ concurrency. We aim to capture the intent of the current ('Final Committee') Draft as closely as possible, but discuss changes that fix many of its problems. We prove that a proposed x86 implementation of the concurrency primitives is correct with respect to the x86-TSO model, and describe our Cppmem tool for exploring the semantics of examples, using code generated from our Isabelle/HOL definitions. Having already motivated changes to the draft standard, this work will aid discussion of any further changes, provide a correctness condition for compilers, and give a much-needed basis for analysis and verification of concurrent C and C++ programs.





## Dissemin

has asked for the following access to your ORCID Record

---



Get your ORCID iD

---

- Allow this permission until I revoke it.  
*You may revoke permissions on your account settings page. Unchecking this box will grant permission this time only.*

This application will not be able to see your ORCID password, or other private info in your ORCID Record. [Privacy Policy](#).

Don't have an ORCID iD? [Register](#).

Email or iD



Password



[Forgotten password!](#)

Deny


## Dépôt de "Mathematizing C++ concurrency"

Vous pouvez déposer le texte intégral de votre article. Dissemin l'enverra à un dépôt où il sera mis à la disposition de tous. En déposant votre article sur Zenodo avec Dissemin, vous acceptez nos [conditions d'utilisation](#).

### Document

Sélectionnez ici le texte intégral de votre article. Seul le PDF est autorisé, taille maximale : 10,0 Mio.

Sélectionnez un fichier :

 Naviguer

Sinon, entrez un URL :



Sinon, glissez-déposez un fichier ici :

### Options

Type de dépôt: **Postdocument** (● dépôt autorisé)

Dépôt : **Zenodo**



Votre document sera archivé sur **Zenodo**, un dépôt ouvert généraliste hébergé par le CERN. Si le document n'a pas encore de DOI, Zenodo en créera un.

Relire les métadonnées

 Dépôt

### Publié dans


Proceedings of the 38th annual ACM SIGPLAN-SIGACT symposium on Principles of programming languages - POPL '11, 2011


DOI: [10.1145/1926385.1926394](https://doi.org/10.1145/1926385.1926394)


ACM SIGPLAN Notices, **1**(46), 2011

DOI: [10.1145/1925844.1926394](https://doi.org/10.1145/1925844.1926394)

### Liens

[\[kar.kent.ac.uk\]](http://kar.kent.ac.uk) 

[Association for Computing Machinery \(ACM\)](#)  via DOI


[Association for Computing Machinery \(ACM\)](#)  via DOI

### Outils

[Chercher dans Google Scholar](#)

[Chercher dans CORE](#)

Sélectionnez un  
fichier :

 Naviguer

Sinon, entrez un URL :

http://



Sinon, glissez-déposez un  
fichier ici :

Options

Type de dépôt: **Postdocument** (● dépôt autorisé)

Dépôt : **Zenodo**

Relire les métadonnées

### Résumé\*

Shared-memory concurrency in C and C++ is pervasive in systems programming, but has long been poorly defined. This motivated an ongoing shared effort by the standards committees to specify concurrent behaviour in the next versions of both languages. They aim to provide strong guarantees for race-free programs, together with new (but subtle) relaxed-memory atomic primitives for high-performance concurrent code. However, the current draft standards, while the result of careful deliberation, are not yet clear and rigorous definitions, and harbour substantial problems in their details. In this paper we establish a mathematical (yet readable) semantics for C++ concurrency. We aim to capture the intent of the current ('Final Committee') Draft as closely as possible, but discuss changes that fix many of its problems. We prove that a proposed x86 implementation of the concurrency primitives is correct with respect to the x86-TSO model, and describe our Cppmem tool for exploring the semantics of examples, using code generated from

### License\*

- Creative Commons CCZero
- Creative Commons Attribution
- Creative Commons Attribution-ShareAlike (Partage à l'identique)
- Creative Commons Attribution-NonCommercial
- Creative Commons Attribution-NoDerivatives (Pas de travaux dérivés)
- Autre (licence non-libre)

 Dépôt

DOI: 10.1145/1925844.1926394

Liens

[\[kar.kent.ac.uk\]](http://kar.kent.ac.uk)

[Association for Computing Machinery \(ACM\)](#) via DOI

[Association for Computing Machinery \(ACM\)](#) via DOI

Outils

[Chercher dans Google Scholar](#)

[Chercher dans CORE](#)

# Mathematizing C++ concurrency

Article dans une revue de Mark Batty, Scott Owens, Susmit Sarkar, Peter Sewell, Tjark Weber

L'article a été correctement déposé !



Texte intégral: [Télécharger](#)

Maison d'édition: Association for Computing Machinery (ACM)

Deposited. [Déposer à nouveau](#)

## Résumé

Shared-memory concurrency in C and C++ is pervasive in systems programming, but has long been poorly defined. This motivated an ongoing shared effort by the standards committees to specify concurrent behaviour in the next versions of both languages. They aim to provide strong guarantees for race-free programs, together with new (but subtle) relaxed-memory atomic primitives for high-performance concurrent code. However, the current draft standards, while the result of careful deliberation, are not yet clear and rigorous definitions, and harbour substantial problems in their details. In this paper we establish a mathematical (yet readable) semantics for C++ concurrency. We aim to capture the intent of the current ('Final Committee') Draft as closely as possible, but discuss changes that fix many of its problems. We prove that a proposed x86 implementation of the concurrency primitives is correct with respect to the x86-TSO model, and describe our Cppmem tool for exploring the semantics of examples, using code generated from our Isabelle/HOL definitions. Having already motivated changes to the draft standard, this work will aid discussion of any further changes, provide a correctness condition for compilers, and give a much-needed basis for analysis and verification of concurrent C and C++ programs.

## Publié dans

Proceedings of the 38th annual ACM SIGPLAN-SIGACT symposium on Principles of programming languages - POPL '11, 2011

DOI: 10.1145/1926385.1926394

ACM SIGPLAN Notices, 1(46), 2011

DOI: 10.1145/1925844.1926394

## Liens

Zenodo [🔗](#) | [📄 PDF](#)

Association for Computing Machinery (ACM) [🔗](#) via DOI

Association for Computing Machinery (ACM) [🔗](#) via DOI

## Outils

[Chercher dans Google Scholar](#)

[Chercher dans CORE](#)

01 January 2011

Journal article Open access

# Mathematizing C++ concurrency

Batty, Mark ; Owens, Scott ; Sarkar, Susmit ; Sewell, Peter ; Weber, Tjark

(show affiliations)

Shared-memory concurrency in C and C++ is pervasive in systems programming, but has long been poorly defined. This motivated an ongoing shared effort by the standards committees to specify concurrent behaviour in the next versions of both languages. They aim to provide strong guarantees for race-free programs, together with new (but subtle) relaxed-memory atomic primitives for high-performance concurrent code. However, the current draft standards, while the result of careful deliberation, are not yet clear and rigorous definitions, and harbour substantial problems in their details. In this paper we establish a mathematical (yet readable) semantics for C++ concurrency. We aim to capture the intent of the current ('Final Committee') Draft as closely as possible, but discuss changes that fix many of its problems. We prove that a proposed x86 implementation of the concurrency primitives is correct with respect to the x86-TSO model, and describe our Cppmem tool for exploring the semantics of examples, using code generated from our Isabelle/HOL definitions. Having already motivated changes to the draft standard, this work will aid discussion of any further changes, provide a correctness condition for compilers, and give a much-needed basis for analysis and verification of concurrent C and C++ programs.

**Publication date:**

01 January 2011

**DOI**[10.1145/1926385.1926394](https://doi.org/10.1145/1926385.1926394)**Published in:**

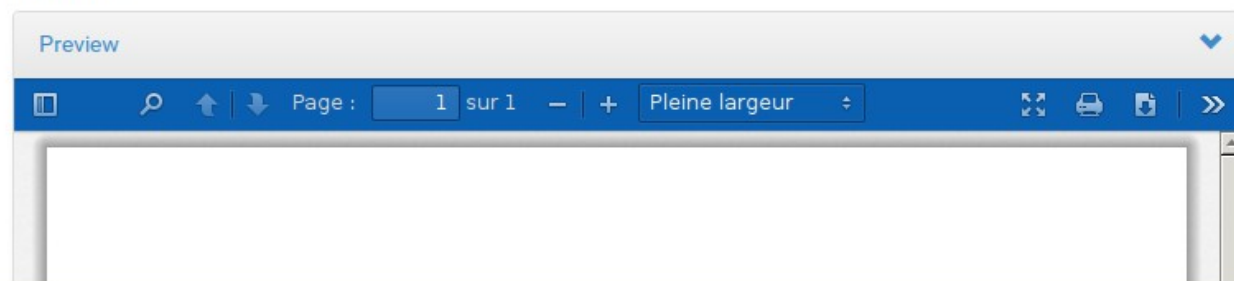
Proceedings of the 38th annual ACM SIGPLAN-SIGACT symposium on Principles of programming languages - POPL '11: (2011)

**License (for files):**[Creative Commons Attribution](#)**Uploaded by:**[dissemin](#) (on 13 October 2015)**New to Zenodo?**

Sign Up

Read more about [features and benefits](#).**Share****Cite as**

Batty, Mark et al. (2011). Mathematizing C++



# Perspectives

Dissemin pour les institutions : faciliter l'analyse des publications de groupes de chercheurs (sans bibliométrie !)

Interfaces vers de nouveaux dépôts (arXiv, PMC...)



equipe@dissem.in - <http://dissem.in/> - @disseminOA

# À vous de jouer !

Nous sommes une association à but non lucratif.

**Vous pouvez nous aider à nous financer.**

La plateforme est ouverte à tous et gratuite.

**Vous pouvez nous aider à nous faire connaître.**

Dissemin est un logiciel libre, modifiable et réutilisable.

**Vous pouvez nous aider à construire Dissemin.**

equipe@dissem.in - <http://dissem.in/> - @disseminOA