

Systèmes de Versionage

introduction, patch & diff

Denys Duchier

IUT d'Orléans

1A OMGL/VCS 2007

- email : `denys.duchier@univ-orleans.fr`
- messagerie instantanée : `denys.duchier@jabber.org`
- page web :
`http://www.univ-orleans.fr/lifo/Members/duchier/`
- il y a une page pour ce cours dans la section Enseignements
- répertoire `/pub/1A/VCS`

Objectifs

- apprendre à utiliser de manière effective les outils d'aide pour :
 - le développement
 - la gestion
 - la maintenance
- de :
 - logiciels
 - documents
 - sites web
 - projets en tous genres
- dans un cadre :
 - collaboratif
 - distribué / délocalisé
 - hétérogène (e.g. contributeurs internes/externes)

Objectifs

- dans ce cours on s'intéressera aux systèmes de versionage
- immédiatement applicable pour vos projets
- nous étudierons 3 générations de systèmes :

système	archivage où ?	archivage quoi ?	quand ?
CVS	centralisé	versionage de fichiers	1989
SVN	centralisé	versionage d'arborescence	2000
BZR	décentralisé	versionage d'arborescence	2005

A quoi sert un VCS ?

- maintenir un historique :
 - qui
 - quoi
 - quand
 - pourquoi

- faciliter la multiplicité :
 - des lieux
 - des gens
 - des tâches/objectifs

Cas 1 : édition d'un document



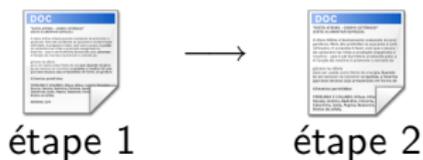
étape 1

- étape 1 : rédaction des sections 1&2
- étape 2 : rédaction de la section 3
- étape 3 : échange des sections 2&3
- étape 4 : rédaction de la section 4

Catastrophe !

un paragraphe initialement dans la section 2 a disparu. Erreur de manœuvre ?

Cas 1 : édition d'un document



- étape 1 : rédaction des sections 1&2
- étape 2 : rédaction de la section 3
- étape 3 : échange des sections 2&3
- étape 4 : rédaction de la section 4

Catastrophe !

un paragraphe initialement dans la section 2 a disparu. Erreur de manœuvre ?

Cas 1 : édition d'un document

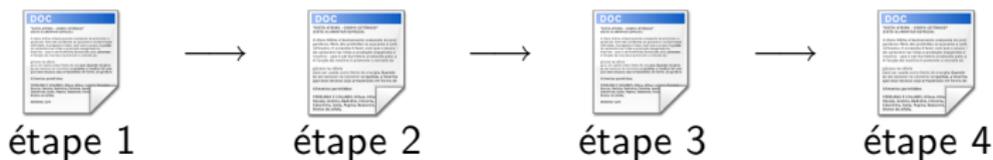


- étape 1 : rédaction des sections 1&2
- étape 2 : rédaction de la section 3
- étape 3 : échange des sections 2&3
- étape 4 : rédaction de la section 4

Catastrophe !

un paragraphe initialement dans la section 2 a disparu. Erreur de manœuvre ?

Cas 1 : édition d'un document

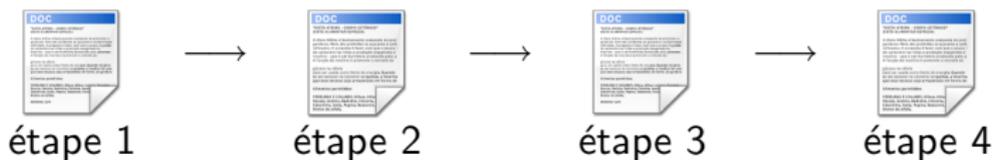


- étape 1 : rédaction des sections 1&2
- étape 2 : rédaction de la section 3
- étape 3 : échange des sections 2&3
- étape 4 : rédaction de la section 4

Catastrophe !

un paragraphe initialement dans la section 2 a disparu. Erreur de manœuvre ?

Cas 1 : édition d'un document



- étape 1 : rédaction des sections 1&2
- étape 2 : rédaction de la section 3
- étape 3 : échange des sections 2&3
- étape 4 : rédaction de la section 4

Catastrophe !

un paragraphe initialement dans la section 2 a disparu. Erreur de manœuvre ?

- comment récupérer le paragraphe perdu ?
- voyager dans le temps pour visiter une version antérieure du document

- comment récupérer le paragraphe perdu ?
- voyager dans le temps pour visiter une version antérieure du document

- comment récupérer le paragraphe perdu ?
- voyager dans le temps pour visiter une version antérieure du document

Version/Révision

une révision d'un document ou d'un projet est un **instantané** pris à un instant donné et sauvegardé dans l'**historique** du projet.

Versions & Historique

- Emacs :
- éditer `foo.txt`
 - sauvegarde `foo.txt~`
 - une seule version dans l'historique

Undo/Redo : `Ctrl+Z` et `Ctrl+Shift+Z`

Versions & Historique

- Emacs :
- éditer `foo.txt`
 - sauvegarde `foo.txt~`
 - une seule version dans l'historique

Undo/Redo : `Ctrl+Z` et `Ctrl+Shift+Z`

Versions & Historique

- Emacs :
- éditer `foo.txt`
 - sauvegarde `foo.txt~`
 - une seule version dans l'historique

Undo/Redo : `Ctrl+Z` et `Ctrl+Shift+Z`

Rôle d'un VCS

la 1ère fonction d'un VCS est de gérer un historique qui permette :

- d'enregistrer de nouvelles révisions à tout moment
- de récupérer n'importe quelle révision enregistrée

Cas 2 : développement d'un logiciel

- vous développez un logiciel
- après plusieurs mois de développement vous vous apercevez que la fonction d'impression ne fonctionne plus
- quand ce bogue a-t-il été introduit ? quelle modification est responsable ?
- solution : réessayer les versions précédentes jusqu'à trouver la dernière sans le bogue
- technique : bisection

Cas 2 : développement d'un logiciel

- vous développez un logiciel
- après plusieurs mois de développement vous vous apercevez que la fonction d'impression ne fonctionne plus
- quand ce bogue a-t-il été introduit ? quelle modification est responsable ?
- solution : réessayer les versions précédentes jusqu'à trouver la dernière sans le bogue
- technique : bisection

Cas 2 : développement d'un logiciel

- vous développez un logiciel
- après plusieurs mois de développement vous vous apercevez que la fonction d'impression ne fonctionne plus
- quand ce bogue a-t-il été introduit ? quelle modification est responsable ?
- solution : réessayer les versions précédentes jusqu'à trouver la dernière sans le bogue
- technique : bisection

Cas 2 : développement d'un logiciel

- vous développez un logiciel
- après plusieurs mois de développement vous vous apercevez que la fonction d'impression ne fonctionne plus
- quand ce bogue a-t-il été introduit ? quelle modification est responsable ?
- solution : réessayer les versions précédentes jusqu'à trouver la dernière sans le bogue
- technique : bisection

Cas 2 : développement d'un logiciel

- vous développez un logiciel
- après plusieurs mois de développement vous vous apercevez que la fonction d'impression ne fonctionne plus
- quand ce bogue a-t-il été introduit ? quelle modification est responsable ?
- solution : réessayer les versions précédentes jusqu'à trouver la dernière sans le bogue
- technique : bisection

Cas 2 : développement d'un logiciel

- vous développez un logiciel
- après plusieurs mois de développement vous vous apercevez que la fonction d'impression ne fonctionne plus
- quand ce bogue a-t-il été introduit ? quelle modification est responsable ?
- solution : réessayer les versions précédentes jusqu'à trouver la dernière sans le bogue
- technique : bisection

Remarque

l'historique est un outil essentiel pour le débogage

Cas 2 : développement d'un logiciel

- vous découvrez ainsi que la dernière version où la fonction d'impression marchait est la version k
- c'est donc la modification effectuée entre la version k et la version $k + 1$ qui est responsable
- mais vous ne vous souvenez plus pourquoi vous l'avez faite
- il faudrait pouvoir documenter chaque modification

Cas 2 : développement d'un logiciel

- vous découvrez ainsi que la dernière version où la fonction d'impression marchait est la version k
- c'est donc la modification effectuée entre la version k et la version $k + 1$ qui est responsable
- mais vous ne vous souvenez plus pourquoi vous l'avez faite
- il faudrait pouvoir documenter chaque modification

Cas 2 : développement d'un logiciel

- vous découvrez ainsi que la dernière version où la fonction d'impression marchait est la version k
- c'est donc la modification effectuée entre la version k et la version $k + 1$ qui est responsable
- mais vous ne vous souvenez plus pourquoi vous l'avez faite
- il faudrait pouvoir documenter chaque modification

Cas 2 : développement d'un logiciel

- vous découvrez ainsi que la dernière version où la fonction d'impression marchait est la version k
- c'est donc la modification effectuée entre la version k et la version $k + 1$ qui est responsable
- mais vous ne vous souvenez plus pourquoi vous l'avez faite
- il faudrait pouvoir documenter chaque modification

Cas 2 : développement d'un logiciel

- vous découvrez ainsi que la dernière version où la fonction d'impression marchait est la version k
- c'est donc la modification effectuée entre la version k et la version $k + 1$ qui est responsable
- mais vous ne vous souvenez plus pourquoi vous l'avez faite
- il faudrait pouvoir documenter chaque modification

Rôle d'un VCS

la 2nd fonction d'un VCS est de documenter chaque **commit** en lui associant un **message de log**

Cas 2 : développement d'un logiciel

- le projet contient une fonction que vous ne comprenez pas : le mieux serait de demander à celui qui l'a programmée
- vous avez localisé une ligne qui semble avoir un bogue, mais vous n'êtes pas sûr. Il faudrait demander l'opinion de celui qui l'a écrite
- il faudrait que l'historique nous permette de déterminer qui a écrit quoi

Cas 2 : développement d'un logiciel

- le projet contient une fonction que vous ne comprenez pas : le mieux serait de demander à celui qui l'a programmée
- vous avez localisé une ligne qui semble avoir un bogue, mais vous n'êtes pas sur. Il faudrait demander l'opinion de celui qui l'a écrite
- il faudrait que l'historique nous permette de déterminer qui a écrit quoi

Cas 2 : développement d'un logiciel

- le projet contient une fonction que vous ne comprenez pas : le mieux serait de demander à celui qui l'a programmée
- vous avez localisé une ligne qui semble avoir un bogue, mais vous n'êtes pas sur. Il faudrait demander l'opinion de celui qui l'a écrite
- il faudrait que l'historique nous permette de déterminer qui a écrit quoi

Cas 2 : développement d'un logiciel

- le projet contient une fonction que vous ne comprenez pas : le mieux serait de demander à celui qui l'a programmée
- vous avez localisé une ligne qui semble avoir un bogue, mais vous n'êtes pas sur. Il faudrait demander l'opinion de celui qui l'a écrite
- il faudrait que l'historique nous permette de déterminer qui a écrit quoi

Rôle d'un VCS

la 3ème fonction d'un VCS est de noter l'**auteur** de chaque commit.

Ceci permet d'associer un responsable à chaque ligne de code dans chaque révision ("annotate", "blame").

Cas 3 : travail multi-postes

- vous travaillez à un projet logiciel
- pendant la journée vous utilisez une station de travail à l'IUT
- le soir, vous travaillez sur votre ordinateur personnel
- comment transférer les modifications de l'un à l'autre ?

Cas 3 : travail multi-postes

- vous travaillez à un projet logiciel
- pendant la journée vous utilisez une station de travail à l'IUT
- le soir, vous travaillez sur votre ordinateur personnel
- comment transférer les modifications de l'un à l'autre ?

Cas 3 : travail multi-postes

- vous travaillez à un projet logiciel
- pendant la journée vous utilisez une station de travail à l'IUT
- le soir, vous travaillez sur votre ordinateur personnel
- comment transférer les modifications de l'un à l'autre ?

Cas 3 : travail multi-postes

- vous travaillez à un projet logiciel
- pendant la journée vous utilisez une station de travail à l'IUT
- le soir, vous travaillez sur votre ordinateur personnel
- comment transférer les modifications de l'un à l'autre ?

Cas 3 : travail multi-postes

- vous travaillez à un projet logiciel
- pendant la journée vous utilisez une station de travail à l'IUT
- le soir, vous travaillez sur votre ordinateur personnel
- comment transférer les modifications de l'un à l'autre ?

Rôle d'un VCS

la 4ème fonction d'un VCS est de faciliter le travail en **multi-postes**.

par exemple, en permettant l'**accès à distance** à un **dépôt partagé** par tous les postes.

Cas 4 : travail en équipe

Dépot



jeu.h



jeu.c

- A et B doivent échanger les modifications qu'ils ont faites
- le VCS sert d'intermédiaire pour cet échange

Cas 4 : travail en équipe

Dépot



jeu.h



jeu.c

A fait une copie de travail



jeu.h



jeu.c

- A et B doivent échanger les modifications qu'ils ont faites
- le VCS sert d'intermédiaire pour cet échange

Cas 4 : travail en équipe

Dépôt



jeu.h



jeu.c

A fait une copie de travail



jeu.h



jeu.c

B fait une copie de travail



jeu.h



jeu.c

- A et B doivent échanger les modifications qu'ils ont faites
- le VCS sert d'intermédiaire pour cet échange

Cas 4 : travail en équipe

Dépot



jeu.h



jeu.c

A modifie jeu.h



jeu.h



jeu.c

B modifie jeu.c



jeu.h



jeu.c

- A et B doivent échanger les modifications qu'ils ont faites
- le VCS sert d'intermédiaire pour cet échange

Cas 4 : travail en équipe

Dépot



jeu.h



jeu.c

A modifie jeu.h



jeu.h



jeu.c

B modifie jeu.c



jeu.h



jeu.c

- A et B doivent échanger les modifications qu'ils ont faites
- le VCS sert d'intermédiaire pour cet échange

Cas 4 : travail en équipe

Dépôt



jeu.h



jeu.c

A modifie jeu.h



jeu.h



jeu.c

B modifie jeu.c



jeu.h



jeu.c

- A et B doivent échanger les modifications qu'ils ont faites
- le VCS sert d'intermédiaire pour cet échange

Cas 4 : travail en équipe

Dépot



jeu.h



jeu.c

Cas 4 : travail en équipe

Dépot



jeu.h



jeu.c

A fait une copie de travail



jeu.h



jeu.c

Cas 4 : travail en équipe

Dépot



jeu.h



jeu.c

A fait une copie de travail



jeu.h



jeu.c

B fait une copie de travail



jeu.h



jeu.c

Cas 4 : travail en équipe

Dépot



jeu.h



jeu.c

A modifie jeu.c



jeu.h



jeu.c

B modifie jeu.c



jeu.h



jeu.c

Cas 4 : travail en équipe

Dépôt



jeu.h



jeu.c

A modifie jeu.c



jeu.h



jeu.c

B modifie jeu.c



jeu.h



jeu.c

- A doit intégrer dans son fichier jeu.c les modifications de B
- B doit intégrer dans son fichier jeu.c les modifications de A
- c'est le problème de la fusion (merge)

Cas 4 : travail en équipe

Dépôt



jeu.h



jeu.c

A modifie jeu.c



jeu.h



jeu.c

B modifie jeu.c



jeu.h



jeu.c

- A doit intégrer dans son fichier jeu.c les modifications de B
- B doit intégrer dans son fichier jeu.c les modifications de A
- c'est le problème de la fusion (merge)

Cas 4 : travail en équipe

Dépôt



jeu.h



jeu.c

A modifie jeu.c



jeu.h



jeu.c

B modifie jeu.c



jeu.h



jeu.c

- A doit intégrer dans son fichier jeu.c les modifications de B
- B doit intégrer dans son fichier jeu.c les modifications de A
- c'est le problème de la **fusion** (merge)

Cas 4 : travail en équipe

Dépot



jeu.h



jeu.c

A modifie jeu.c



jeu.h



jeu.c

B modifie jeu.c



jeu.h



jeu.c

Rôle d'un VCS

la 5ème fonction d'un VCS est de faciliter la **fusion de modifications** effectuées par différents collaborateurs et de **gérer les conflits**

Cas 5 : un même logiciel / différentes tâches

- **plusieurs systèmes d'exploitation** : Linux & Windows & MacOS
- plusieurs lignes :
 - produit commercialisé (stable, en maintenance)
 - nouvelle idée (instable, en développement)
- plusieurs cibles :
 - logiciel bridé (version de démonstration)
 - logiciel "home edition"
 - logiciel "enterprise edition"
- plusieurs bogues :
 - logiciel commercialisé
 - réparation du bogue 277
 - réparation du bogue 389

Cas 5 : un même logiciel / différentes tâches

- plusieurs systèmes d'exploitation : Linux & Windows & MacOS
- plusieurs lignes :
 - produit commercialisé (stable, en maintenance)
 - nouvelle idée (instable, en développement)
- plusieurs cibles :
 - logiciel bridé (version de démonstration)
 - logiciel "home edition"
 - logiciel "enterprise edition"
- plusieurs bogues :
 - logiciel commercialisé
 - réparation du bogue 277
 - réparation du bogue 389

Cas 5 : un même logiciel / différentes tâches

- plusieurs systèmes d'exploitation : Linux & Windows & MacOS
- plusieurs lignes :
 - produit commercialisé (stable, en maintenance)
 - nouvelle idée (instable, en développement)
- **plusieurs cibles** :
 - logiciel bridé (version de démonstration)
 - logiciel “home edition”
 - logiciel “enterprise edition”
- plusieurs bogues :
 - logiciel commercialisé
 - réparation du bogue 277
 - réparation du bogue 389

Cas 5 : un même logiciel / différentes tâches

- plusieurs systèmes d'exploitation : Linux & Windows & MacOS
- plusieurs lignes :
 - produit commercialisé (stable, en maintenance)
 - nouvelle idée (instable, en développement)
- plusieurs cibles :
 - logiciel bridé (version de démonstration)
 - logiciel “home edition”
 - logiciel “enterprise edition”
- plusieurs bogues :
 - logiciel commercialisé
 - réparation du bogue 277
 - réparation du bogue 389

Cas 5 : un même logiciel / différentes tâches

- plusieurs systèmes d'exploitation : Linux & Windows & MacOS
- plusieurs lignes :
 - produit commercialisé (stable, en maintenance)
 - nouvelle idée (instable, en développement)
- plusieurs cibles :
 - logiciel bridé (version de démonstration)
 - logiciel "home edition"
 - logiciel "enterprise edition"
- plusieurs bogues :
 - logiciel commercialisé
 - réparation du bogue 277
 - réparation du bogue 389

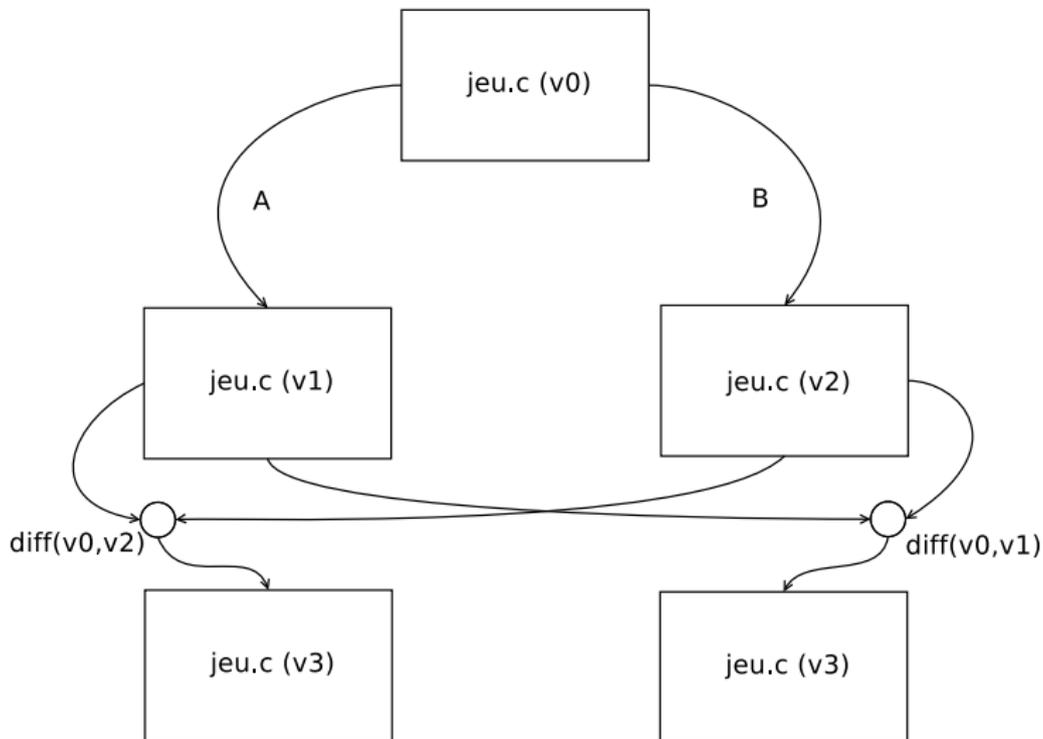
Rôle d'un VCS

la 6ème fonction d'un VCS est de faciliter le développement parallèle de multiples **branches** et le **transfert de modifications** entre branches.

- maintenir un historique :
 - qui
 - quoi
 - quand
 - pourquoi

- faciliter la multiplicité :
 - des lieux
 - des gens
 - des tâches/objectifs

Fusion : notion de delta/diff



Fusion : notion de delta/diff

- fusion : on a besoin d'une notion de modification plus fine que le simple remplacement d'une version par une autre
- A doit donner à B juste la **différence** correspondant à sa mod pour que B puisse l'appliquer à sa copie
- une différence s'appelle aussi un **delta** ou un **diff**
- $\text{diff}(F1, F2)$ est le delta entre F1 et F2
- `patch` permet d'appliquer un delta

Propriété essentielle

$$\text{patch}(F1, \text{diff}(F1, F2)) = F2$$

Fusion : notion de delta/diff

- supposons que A et B on chacun obtenu une copie de F0
- A l'a modifié produisant F1
- B l'a modifié produisant F2
- pour que la notion de delta soit utile, il faudrait qu'un delta $\text{diff}(F0,F1)$ produit par A puisse être utilisé par B pour mettre sa copie F2 à jour et réciproquement

Propriété désirée

$$\text{patch}(F2,\text{diff}(F0,F1)) = \text{patch}(F1,\text{diff}(F0,F2))$$

F0

A
B
C
D

F1

C
C2
D

```
diff -u F0 F1
```

```
--- F0 2007-11-07 00:12:31.000000000 +0100
```

```
+++ F1 2007-11-07 00:12:47.000000000 +0100
```

```
@@ -1,4 +1,3 @@
```

```
-A
```

```
-B
```

```
C
```

```
+C2
```

```
D
```

- un diff est un programme
- il représente :
 - comment modifier le fichier source
 - pour obtenir le fichier cible
- ce programme est calculé par `diff`
- ce programme est interprété par `patch`
- les lignes de `contexte` permettent d'appliquer le delta à des fichiers autres (mais assez proches) du fichier source

diff & patch : example

D1/F

```
A
B
C
D
```

D2/F

```
C
C2
D
```

D3/F

```
Z
A
B
C
U
```

```
cd D3; diff -u ../D1/F ../D2/F
```

```
--- ../D1/F 2007-11-07 13:01:44.000000000 +0100
```

```
+++ ../D2/F 2007-11-07 13:10:13.000000000 +0100
```

```
@@ -1,4 +1,3 @@
```

```
-A
```

```
-B
```

```
 C
```

```
+C2
```

```
 D
```

diff & patch : example

D1/F

```
A
B
C
D
```

D2/F

```
C
C2
D
```

D3/F

```
Z
A
B
C
U
```

```
cd D3; diff -u ../D1/F ../D2/F | patch
```

```
patching file F
```

```
Hunk #1 succeeded at 2 with fuzz 1 (offset 1 line).
```

diff & patch : example

D1/F

```
A
B
C
D
```

D2/F

```
C
C2
D
```

D3/F

```
Z
C
C2
U
```

```
cd D3; diff -u ../D1/F ../D2/F | patch
```

```
patching file F
```

```
Hunk #1 succeeded at 2 with fuzz 1 (offset 1 line).
```

dépôt

A
B
C

moi

~~A~~
~~B~~
C

lui

A
B
D
B
C

- comment transférer les modifications de "moi" dans "lui" ?

- $\text{diff}(\text{moi}, \text{lui}) = \text{insérer}$

A
B
D
B

avant C

- ce n'est pas ce qu'on veut

dépôt

A
B
C

moi

~~A~~
~~B~~
C

lui

A
B
D
B
C

- comment transférer les modifications de "moi" dans "lui" ?

- $\text{diff}(\text{moi}, \text{lui}) = \text{insérer}$  avant C

- ce n'est pas ce qu'on veut

dépôt

A
B
C

moi

~~A~~
~~B~~
C

lui

A
B
D
B
C

- comment transférer les modifications de "moi" dans "lui" ?

- $\text{diff}(\text{moi}, \text{lui}) = \text{insérer}$

A
B
D
B

avant C

- ce n'est pas ce qu'on veut

dépôt

A
B
C

moi

~~A~~
~~B~~
C

lui

A
B
D
B
C

- comment transférer les modifications de “moi” dans “lui” ?

- $\text{diff}(\text{moi}, \text{lui}) = \text{insérer}$

A
B
D
B

avant C

- ce n'est pas ce qu'on veut

dépôt

A
B
C

moi

~~A~~
~~B~~
C

lui

A
B
D
B
C

- “moi” supprime les 2 premières lignes
- “lui” insère 2 lignes avant C
- ces deux actions ne sont pas en conflit

dépôt

A
B
C

moi

~~A~~
~~B~~
C

lui

A
B
D
B
C

- “moi” supprime les 2 premières lignes
- “lui” insère 2 lignes avant C
- ces deux actions ne sont pas en conflit

dépôt

A
B
C

moi

~~A~~
~~B~~
C

lui

A
B
D
B
C

- “moi” supprime les 2 premières lignes
- “lui” insère 2 lignes avant C
- ces deux actions ne sont pas en conflit

dépôt

A
B
C

moi

~~A~~
~~B~~
C

lui

A
B
D
B
C

- $\text{diff}(\text{dépôt}, \text{moi}) = \text{supprime}$

A
B
- $\text{diff}(\text{dépôt}, \text{lui}) = \text{ajoute}$

D
B

 avant C
- on peut effectuer ces deux opérations sur la version du dépôt

dépôt

A
B
C

moi

~~A~~
~~B~~
C

lui

A
B
D
B
C

- $\text{diff}(\text{dépôt}, \text{moi}) = \text{supprime}$

A
B
- $\text{diff}(\text{dépôt}, \text{lui}) = \text{ajoute}$

D
B

 avant C
- on peut effectuer ces deux opérations sur la version du dépôt

dépôt

A
B
C

moi

~~A~~
~~B~~
C

lui

A
B
D
B
C

- $\text{diff}(\text{dépôt}, \text{moi}) = \text{supprime}$

A
B
- $\text{diff}(\text{dépôt}, \text{lui}) = \text{ajoute}$

D
B

 avant C
- on peut effectuer ces deux opérations sur la version du dépôt

A
B
C

~~A~~
~~B~~
C

~~A~~
~~B~~
D
B
C

- fusion tri-partite (3-way merge)
- base + 2 versions
- `kdiff3`

A
B
C

~~A~~
~~B~~
C

~~A~~
~~B~~
D
B
C

- fusion tri-partite (3-way merge)
- base + 2 versions
- `kdiff3`

A
B
C

~~A~~
~~B~~
C

~~A~~
~~B~~
D
B
C

- fusion tri-partite (3-way merge)
- base + 2 versions
- `kdiff3`

A
B
C

~~A~~
~~B~~
C

~~A~~
~~B~~
D
B
C

- fusion tri-partite (3-way merge)
- base + 2 versions
- `kdiff3`

Ambiguïté de fusion

dépôt

A
B

moi

A
C
B

lui

A
D
B

résultat ?

A
C
D
B

A
D
C
B

Stockage différentiel

- supposons que l'historique enregistre chaque version telle quelle
- Proust (7v) ou Chateaubriand (42v) vont rapidement remplir le disque
- stocker les différences entre révisions plutôt que les révisions elles-mêmes :

v0

- supposons que l'historique enregistre chaque version telle quelle
- Proust (7v) ou Chateaubriand (42v) vont rapidement remplir le disque
- stocker les différences entre révisions plutôt que les révisions elles-mêmes :

v0

Stockage différentiel

- supposons que l'historique enregistre chaque version telle quelle
- Proust (7v) ou Chateaubriand (42v) vont rapidement remplir le disque
- stocker les différences entre révisions plutôt que les révisions elles-mêmes :

v0

Stockage différentiel

- supposons que l'historique enregistre chaque version telle quelle
- Proust (7v) ou Chateaubriand (42v) vont rapidement remplir le disque
- stocker les différences entre révisions plutôt que les révisions elles-mêmes :

v0

Stockage différentiel

- supposons que l'historique enregistre chaque version telle quelle
- Proust (7v) ou Chateaubriand (42v) vont rapidement remplir le disque
- stocker les différences entre révisions plutôt que les révisions elles-mêmes :



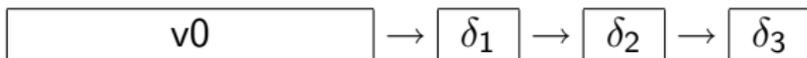
Stockage différentiel

- supposons que l'historique enregistre chaque version telle quelle
- Proust (7v) ou Chateaubriand (42v) vont rapidement remplir le disque
- stocker les différences entre révisions plutôt que les révisions elles-mêmes :



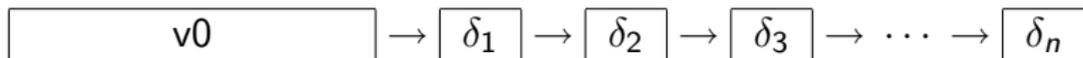
Stockage différentiel

- supposons que l'historique enregistre chaque version telle quelle
- Proust (7v) ou Chateaubriand (42v) vont rapidement remplir le disque
- stocker les différences entre révisions plutôt que les révisions elles-mêmes :



Stockage différentiel

- supposons que l'historique enregistre chaque version telle quelle
- Proust (7v) ou Chateaubriand (42v) vont rapidement remplir le disque
- stocker les différences entre révisions plutôt que les révisions elles-mêmes :



Généralisation : “edit script”

- compression différentielle (WebDAV, deltaV)
- instructions :
 - COPY(longueur, index)
 - ADD(longueur, octets)
- exemple :
 - R = ABCDEFGHIJKLMNOP
 - V = QWIKLNMNOBCDEFGHZDEFGHIJKL
- edit script :

```
ADD(2, QW) COPY(7, 8) COPY(7, 1) ADD(1, Z) COPY(9, 3)
```

Généralisation : “edit script”

- compression différentielle (WebDAV, deltaV)
- instructions :
 - COPY(longueur, index)
 - ADD(longueur, octets)
- exemple :
 - R = ABCDEFGHIJKLMNOP
 - V = QWIKLNMNOBCDEFGHZDEFGHIJKL
- edit script :
ADD(2, QW) COPY(7, 8) COPY(7, 1) ADD(1, Z) COPY(9, 3)

Généralisation : “edit script”

- compression différentielle (WebDAV, deltaV)
- instructions :
 - COPY(longueur, index)
 - ADD(longueur, octets)
- exemple :
 - R = ABCDEFGHIJKLMNOP
 - V = QWIKLNMNOBCDEFGHZDEFGHIJKL
- edit script :

ADD(2, QW) COPY(7, 8) COPY(7, 1) ADD(1, Z) COPY(9, 3)

Généralisation : “edit script”

- compression différentielle (WebDAV, deltaV)
- instructions :
 - COPY(longueur, index)
 - ADD(longueur, octets)
- exemple :
 - R = ABCDEFGHIJKLMNOP
 - V = QWIKLNMNOBCDEFGHZDEFGHIJKL
- edit script :

```
ADD(2, QW) COPY(7, 8) COPY(7, 1) ADD(1, Z) COPY(9, 3)
```

Généralisation : “edit script”

- compression différentielle (WebDAV, deltaV)
- instructions :
 - COPY(longueur, index)
 - ADD(longueur, octets)
- exemple :
 - R = ABCDEFGHIJKLMNOP
 - V = QW IJKLMNOPBCDEFGHZDEFGHIJKL
- edit script :
ADD(2, QW) COPY(7, 8) COPY(7, 1) ADD(1, Z) COPY(9, 3)

Généralisation : “edit script”

- compression différentielle (WebDAV, deltaV)
- instructions :
 - COPY(longueur, index)
 - ADD(longueur, octets)
- exemple :
 - R = ABCDEFGHIJKLMNOP
 - V = QWIKLMNOBCDEFGHZDEFGHIJKL
- edit script :
ADD(2, QW) COPY(7, 8) COPY(7, 1) ADD(1, Z) COPY(9, 3)

Généralisation : “edit script”

- compression différentielle (WebDAV, deltaV)
- instructions :
 - COPY(longueur, index)
 - ADD(longueur, octets)
- exemple :
 - R = ABCDEF**GH**IJKLMNOP
 - V = QWIKLMNO**BCDEF**GHZDEFGHIJKL
- edit script :
ADD(2, QW) COPY(7, 8) COPY(7, 1) ADD(1, Z) COPY(9, 3)

Généralisation : “edit script”

- compression différentielle (WebDAV, deltaV)
- instructions :
 - COPY(longueur, index)
 - ADD(longueur, octets)
- exemple :
 - R = ABCDEFGHIJKLMNOP
 - V = QWIKLNMNOBCDEFGH**Z**DEFGHIJKL
- edit script :
ADD(2, QW) COPY(7, 8) COPY(7, 1) ADD(1, Z) COPY(9, 3)

Généralisation : “edit script”

- compression différentielle (WebDAV, deltaV)
- instructions :
 - COPY(longueur, index)
 - ADD(longueur, octets)
- exemple :
 - R = ABCDEF**FGHI**JKLMNOP
 - V = QWIKLMNOBCDEFGHZ**DEFGHIJKL**
- edit script :
ADD(2, QW) COPY(7, 8) COPY(7, 1) ADD(1, Z) COPY(9, 3)

- **codecs : comment compresser les films (DVD)**
- HDTV : haute définition et bande passante limitée
- chaque image est hautement corrélée avec la précédente
- comment calculer l'image suivante à partir de la précédente ?
- “compension de déplacement”
- déplacements de blocs
- MPEG-4

- codecs : comment compresser les films (DVD)
- HDTV : haute définition et bande passante limitée
- chaque image est hautement corrélée avec la précédente
- comment calculer l'image suivante à partir de la précédente ?
- “compensation de déplacement”
- déplacements de blocs
- MPEG-4

- codecs : comment compresser les films (DVD)
- HDTV : haute définition et bande passante limitée
- chaque image est hautement corrélée avec la précédente
- comment calculer l'image suivante à partir de la précédente ?
- “compensation de déplacement”
- déplacements de blocs
- MPEG-4

- codecs : comment compresser les films (DVD)
- HDTV : haute définition et bande passante limitée
- chaque image est hautement corrélée avec la précédente
- comment calculer l'image suivante à partir de la précédente ?
- “compension de déplacement”
- déplacements de blocs
- MPEG-4

Compression différentielle : vidéo

- codecs : comment compresser les films (DVD)
- HDTV : haute définition et bande passante limitée
- chaque image est hautement corrélée avec la précédente
- comment calculer l'image suivante à partir de la précédente ?
- “compensation de déplacement”
- déplacements de blocs
- MPEG-4

- codecs : comment compresser les films (DVD)
- HDTV : haute définition et bande passante limitée
- chaque image est hautement corrélée avec la précédente
- comment calculer l'image suivante à partir de la précédente ?
- “compensation de déplacement”
- déplacements de blocs
- MPEG-4

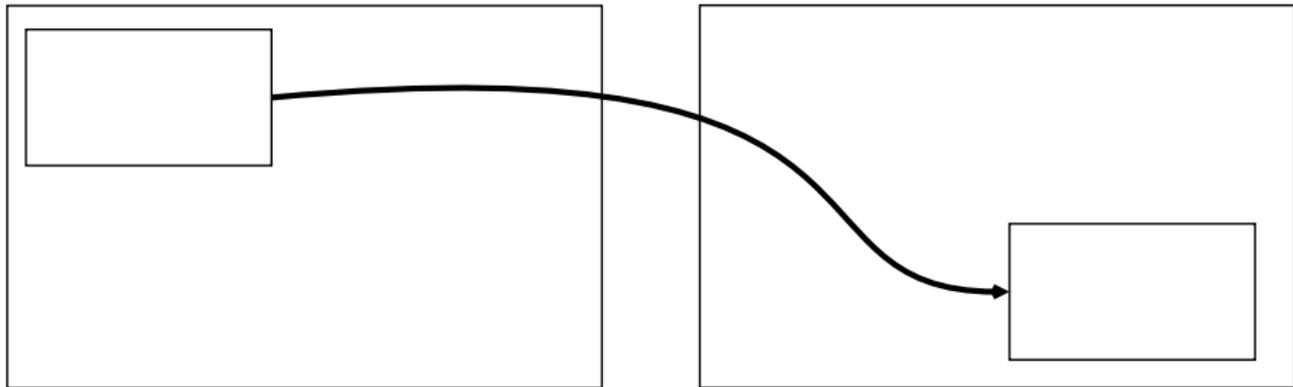
Compression différentielle : vidéo

- codecs : comment compresser les films (DVD)
- HDTV : haute définition et bande passante limitée
- chaque image est hautement corrélée avec la précédente
- comment calculer l'image suivante à partir de la précédente ?
- “compensation de déplacement”
- déplacements de blocs
- MPEG-4

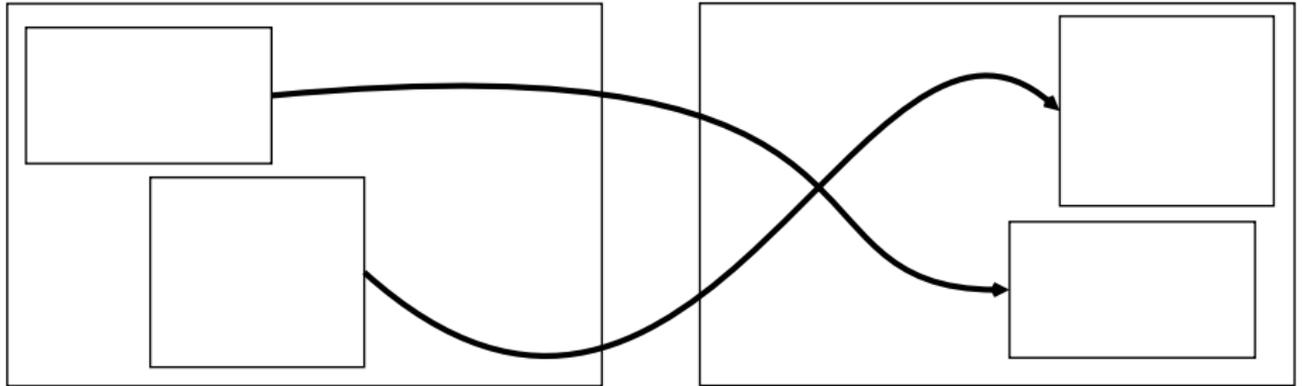
Compression différentielle : vidéo



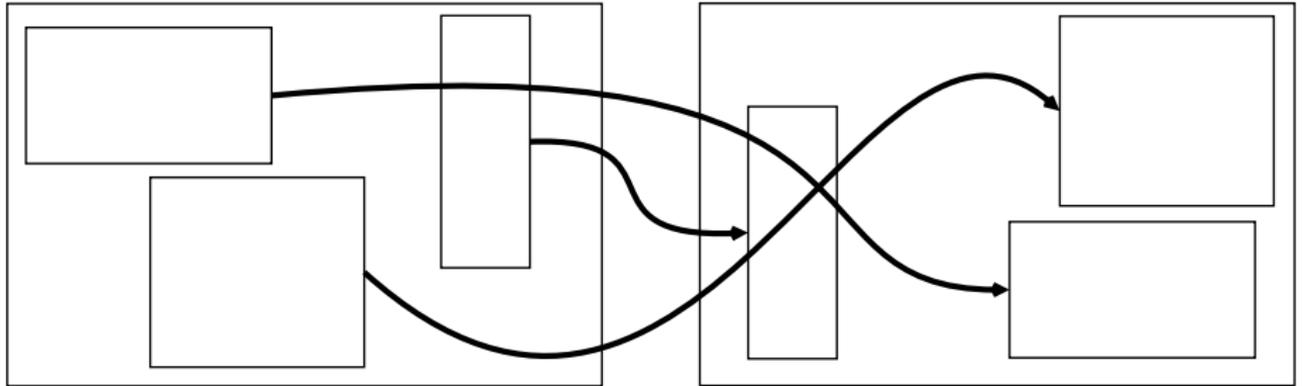
Compression différentielle : vidéo



Compression différentielle : vidéo



Compression différentielle : vidéo



- plutôt que de compresser par rapport à une version précédente
- compresser le suffixe par rapport au préfixe :
- on peut limiter à une fenêtre
- on peut optimiser les instructions

Compression différentielle : document

- plutôt que de compresser par rapport à une version précédente
- compresser le suffixe par rapport au préfixe :



- on peut limiter à une fenêtre
- on peut optimiser les instructions

Compression différentielle : document

- plutôt que de compresser par rapport à une version précédente
- compresser le suffixe par rapport au préfixe :



- on peut limiter à une fenêtre
- on peut optimiser les instructions

Compression différentielle : document

- plutôt que de compresser par rapport à une version précédente
- compresser le suffixe par rapport au préfixe :



- on peut limiter à une fenêtre
- on peut optimiser les instructions