

Introduction à GNUstep

Guillaume Libersat

1^{er} mars 2005

GNUstep en quelques mots

De quoi s'agit-il ?

- ▶ Environnement de développement
- ▶ Orienté logiciel de bureau

GNUstep en quelques mots

De quoi s'agit-il ?

- ▶ Environnement de développement
- ▶ Orienté logiciel de bureau

Caractéristiques

- ▶ Orienté-objet
- ▶ Standard
- ▶ Portable
- ▶ Éprouvé
- ▶ Libre

Que fournit GNUstep ?

GNUstep fournit pour le développement

- ▶ Des librairies
- ▶ Des outils
- ▶ Des applications

Que fournit GNUstep ?

GNUstep fournit pour le développement

- ▶ Des librairies
- ▶ Des outils
- ▶ Des applications

Les applications en résultant sont

- ▶ Rapides
- ▶ Fiables
- ▶ Portables

Un peu d'histoire

Son ascendance...

Un peu d'histoire

Son ascendance...

- ▶ Implémente OpenStep
 - ▶ Créé par NeXT et Sun
 - ▶ Réalisé en 1994

Un peu d'histoire

Son ascendance...

- ▶ Implémente OpenStep
 - ▶ Créé par NeXT et Sun
 - ▶ Réalisé en 1994
- ▶ OpenStep se base sur NeXTSTEP
 - ▶ Créé par NeXT
 - ▶ Utilisable vers la fin des années 80

Un peu d'histoire

Son ascendance...

- ▶ Implémente OpenStep
 - ▶ Créé par NeXT et Sun
 - ▶ Réalisé en 1994
- ▶ OpenStep se base sur NeXTSTEP
 - ▶ Créé par NeXT
 - ▶ Utilisable vers la fin des années 80

De nos jours...

- ▶ GNUstep est compatible Cocoa (MacOSX)
- ▶ GNUstep est compatible avec tout OpenStep

Développer avec GNUstep

Langages

- ▶ Écrit en Objective-C
- ▶ Des wrappers existent vers

Développer avec GNUstep

Langages

- ▶ Écrit en Objective-C
- ▶ Des wrappers existent vers
 - ▶ Java (JIGS)
 - ▶ Ruby (RIGS)

Développer avec GNUstep

Langages

- ▶ Écrit en Objective-C
- ▶ Des wrappers existent vers
 - ▶ Java (JIGS)
 - ▶ Ruby (RIGS)

Librairie

- ▶ Complète
- ▶ Documentée

Développer avec GNUstep

Langages

- ▶ Écrit en Objective-C
- ▶ Des wrappers existent vers
 - ▶ Java (JIGS)
 - ▶ Ruby (RIGS)

Librairie

- ▶ Complète
- ▶ Documentée
- ▶ Gère de nombreuses choses
 - ▶ Chaînes, Tableaux, Threads...
 - ▶ Mémoire (Collecteur)
 - ▶ Objets distribués

Développer avec GNUstep

Langages

- ▶ Écrit en Objective-C
- ▶ Des wrappers existent vers
 - ▶ Java (JIGS)
 - ▶ Ruby (RIGS)

Librairie

- ▶ Complète
- ▶ Documentée
- ▶ Gère de nombreuses choses
 - ▶ Chaînes, Tableaux, Threads...
 - ▶ Mémoire (Collecteur)
 - ▶ Objets distribués

Applications

- ▶ ProjectCenter (Clone de ProjectBuilder)
- ▶ Gorm (Graphical Object Relationship Modeller)

Interlude : Objective-C

De quoi s'agit-il ?

- ▶ Langage objet
- ▶ Écrit au dessus du C
- ▶ Utilise la syntaxe infixée du SmallTalk

Interlude : Objective-C

De quoi s'agit-il ?

- ▶ Langage objet
- ▶ Écrit au dessus du C
- ▶ Utilise la syntaxe infixée du SmallTalk

Fonctions clés

- ▶ Typages dynamiques
- ▶ Appels dynamiques
- ▶ Simple
- ▶ Élégant
- ▶ Rapide

Objective-C : Exemple

Exemple (Création d'objet)

id unObjet = [Voiture new]

ou

id unObjet = [[Voiture alloc] init]

Objective-C : Exemple

Exemple (Création d'objet)

```
id unObjet = [Voiture new]
```

ou

```
id unObjet = [[Voiture alloc] init]
```

Exemple (Utilisation du Collecteur)

```
id unObjet = AUTORELEASE([Voiture new]);
```

ou

```
id unObjet = [[Voiture new] autorelease];
```

Objective-C : Exemple

Exemple (Création d'objet)

```
id unObjet = [Voiture new]
```

ou

```
id unObjet = [[Voiture alloc] init]
```

Exemple (Utilisation du Collecteur)

```
id unObjet = AUTORELEASE([Voiture new]);
```

ou

```
id unObjet = [[Voiture new] autorelease];
```

Exemple (Appel de méthode)

```
[myColor setRed :0.1 green :1.3 blue :0.5];
```

Une classe Objective-C

Exemple (Déclaration d'interface)

```
#include "Foundation/Foundation.h"
@interface Member : NSObject
{
    char *name;
    char *firstname;
    Member *next;
}
- (Member*)newMember;
- (void)setFirstName :(char*)newFirstName;
@end
```

Une classe Objective-C

Exemple (Implémentation)

```
#include "Member.h"
@implementation Member
- (Member*)newMember
{
    self->next = [Member new];
    return self->next;
}
- (void)setFirstName :(char*)NewFirstName
{
    if ( self->firstname != NULL )
    ...
}
@end
```

Développer avec GNUstep

La base

- ▶ Contenue dans “FoundationKit”
- ▶ Toutes les classes commencent par “NS”
- ▶ Concerne tout ce qui n'est pas graphique
- ▶ Tout fichier doit inclure “Foundation/Foundation.h”
- ▶ Toute classe doit hériter de “NSObject”

Développer avec GNUstep

La base

- ▶ Contenue dans “FoundationKit”
- ▶ Toutes les classes commencent par “NS”
- ▶ Concerne tout ce qui n'est pas graphique
- ▶ Tout fichier doit inclure “Foundation/Foundation.h”
- ▶ Toute classe doit hériter de “NSObject”

Le graphique

- ▶ Programmé dans “AppKit”
- ▶ Utilisation de Gorm pour créer l'interface
 - ▶ Extrêmement proche de la programmation
 - ▶ Très facile avec son abstraction
- ▶ Interface “vieillote”

GNUstep Frameworks

Quelques frameworks :

- ▶ GDL2 : Bases de données
- ▶ GNUstepWeb : Développement web de type MVC (compatible WebObjects 4.x)
- ▶ MusicKit : Système de création, édition sonore orienté objet
- ▶ 3DKit : une passerelle vers OpenGL
- ▶ CameraKit : Librairie permettant l'accès aux appareils photo (avec la libgphoto)

Environnements de développement

ProjectCenter

- ▶ IDE fait pour GNUstep
- ▶ Calqué sur ProjectBuilder
- ▶ Simple à utiliser
- ▶ Travaille avec Gorm

Environnements de développement

ProjectCenter

- ▶ IDE fait pour GNUstep
- ▶ Calqué sur ProjectBuilder
- ▶ Simple à utiliser
- ▶ Travaille avec Gorm

Gorm

- ▶ Créateur d'interfaces
- ▶ Facile et performant
- ▶ Très rapide

Makefiles et structure

Construction

- ▶ Fichier “Makefile” de type GNU, appelé “GNUmakefile”
 - ▶ Très simple (2 lignes de base)
 - ▶ Réutilise un Makefile générique de GNUstep
- ▶ Pas de présence d'un “configure”

Makefiles et structure

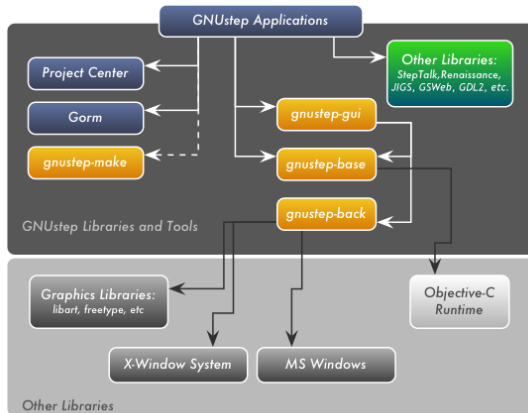
Construction

- ▶ Fichier “Makefile” de type GNU, appelé “GNUmakefile”
 - ▶ Très simple (2 lignes de base)
 - ▶ Réutilise un Makefile générique de GNUstep
- ▶ Pas de présence d’un “configure”

Fichiers de sortie

- ▶ Répertoire “Nom.app”
- ▶ Contient le binaire “Nom”
- ▶ Contient un dossier “Ressources”
 - ▶ Contient divers fichiers, comme l’icone pour le bureau
- ▶ un “.app” s’exécute nativement, avec un double-click

Schéma de l'architecture



Construisons un multiplicateur

- ▶ Interface graphique
- ▶ Deux champs de saisie
- ▶ Un champ de résultat
- ▶ Un bouton pour effectuer le calcul
- ▶ Et le moins de code possible !

Plus d'informations

Pour plus d'informations :

- ▶ GNUstep : <http://www.gnustep.org>
- ▶ Wiki GNUstep : <http://mediawiki.gnustep.org>
- ▶ GNUstepWeb : <http://www.gnustepweb.org>
- ▶ Documentations : <http://www.roard.com/docs/>
- ▶ Camaleon : <http://www.roard.com/camaeleon/>

Plus d'informations

Pour plus d'informations :

- ▶ GNUstep : <http://www.gnustep.org>
- ▶ Wiki GNUstep : <http://mediawiki.gnustep.org>
- ▶ GNUstepWeb : <http://www.gnustepweb.org>
- ▶ Documentations : <http://www.roard.com/docs/>
- ▶ Camaleon : <http://www.roard.com/camaeleon/>

Des questions ?