

# Environnements et Outils de Développement

## Cours 1 — Introduction

Stefano Zacchioli  
zack@pps.univ-paris-diderot.fr

Laboratoire PPS, Université Paris Diderot

2012-2013

URL <http://upsilon.cc/zack/teaching/1213/ed6/>  
Copyright © 2012-2013 Stefano Zacchioli  
License Creative Commons Attribution-ShareAlike 3.0 Unported License  
<http://creativecommons.org/licenses/by-sa/3.0/>



# “Développer” ...

... une activité complexe, formée par plusieurs sous-activités :

## Développement de logiciel

le travail de :

- étudier
- concevoir
- construire
- déboguer
- paramétrer
- documenter
- maintenir
- installer
- mettre à jour
- améliorer
- ...

des logiciels

# Outils de développement

Le **génie logiciel** est la branche de l'informatique et du génie industriel qui s'occupe des **méthodes des travail** et des **bonnes pratiques** autour des activités du développement.

L'évolution de **complexité** des tâches et des langages de programmation a exigé la création des logiciels pour le **traitement (semi-)automatique de programmes** pendant leur développement.

## Outil de développement

Un **outil de développement** est un logiciel qui aide un développeur dans le déroulement d'une activité de développement.

# L'importance des outils de développement

Dans le cas général, les outils de développement nous aident à :

- ❶ implanter une phase d'un **processus de développement logiciel**
- ❷ automatiser des tâches importantes *et* ennuyeux
- ❸ être plus efficace

- sans (1), nous ne pourrions pas avancer dans le développement
  - ▶ p.ex. comment écrire un programme sans un éditeur (de texte)?
  - ▶ comment l'exécuter sans un compilateur ou un interprète?

# L'importance des outils de développement (cont.)

Dans le cas général, les outils de développement nous aident à :

- ❶ implanter une phase d'un **processus de développement logiciel**
  - ❷ automatiser des tâches importantes *et* ennuyeux
  - ❸ être plus efficace
- sans (2) et (3), notre temps serait occupés par des tâches moins “nobles” que la **conception abstraite**, l'**algorithmique**, la **résolution de problèmes** — qui constituent les vrais **habilités du développeurs**
    - ▶ p.ex. combien de **temps** il vous faut pour (re-)indenter une fonction de 40 lignes ?
    - ▶ et pour renommer (sans capture) une structure de données dans 20 fichiers source ?
    - ▶ pour exécuter 99 tests unitaires après un *bug fix* ?
    - ▶ déployer toto 2.0 sur 1'000 machines ?
    - ▶ combien d'**actions manuelles** dans chaque cas ?

# Outils de développement

La pratique du génie logiciel à travers des années nous a amenés à l'utilisation d'une multitude des outils de développement.

- édition du code
- compilation
- débogage
- analyse des dépendances
- génération de doc.
- tester
- archiver
- publier
- analyse d'empreinte mémoire
- analyse les performances
- automatisation des taches
- gestion des différences
- gestion des versions
- gestion de paquets
- ...

# Objectif du cours

Dans votre vie de développeur, l'utilisation des outils correspondantes à toute tâches du développement logiciel sera quotidienne. Les maîtriser est impératif (et dans votre intérêt).

## Objectif du cours

### Maîtriser les outils du développement logiciel

- **efficacité** dans l'exécution de tâche fréquent, non automatisables
  - ▶ *si vous passez une heure à apprendre comment gagner 20 seconds sur une tâche que vous répétez 50 fois par jour, en combien de jours vous aurez plus de temps à disposition ?*
- **automatisation** de tâches répétitives
  - ▶ même gagne de temps qu'avant, car l'ordinateur est souvent beaucoup plus rapide que nous !
  - ▶ plus d'automatisation → moins d'erreurs

(avec quelle assumption ?)

Des **outils fondamentaux**—qui font une seule chose, bien—aux **outils complexes**.

- maîtriser les composantes individuelles, pour mieux comprendre leur interactions et pouvoir en suite maîtriser leur **agrégations**
- le paradigme des **environnements de développement intégré** (IDE) comme orchestrations d'outils plus simple

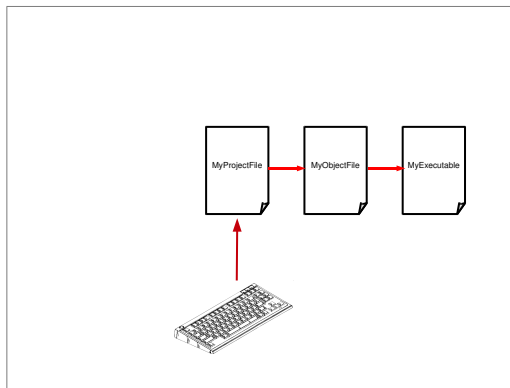


Accent sur les outils de développement typiques du **logiciel libre et open source** (FOSS) (et FOSS eux même)

- avantage didactique : on peut étudier leur fonctionnement
- une grosse partie des activités de développement sont liée à la **collaboration entre développeurs** ; le monde du libre est un cas extrême de collaboration
- demande importante et à la hausse dans le marché IT



# Plan du cours — compilation

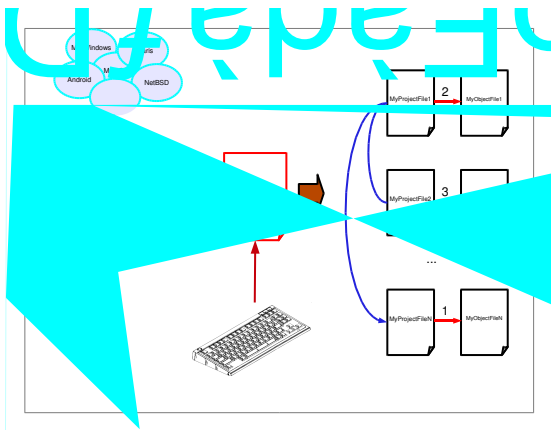


- la chaîne de compilation
  - ▶ pré-processeur
  - ▶ compilateur
  - ▶ éditeur de liens
  - ▶ assembleur
- liaison statique
- liaison dynamique

Outils : gcc



# Plan du cours — configuration

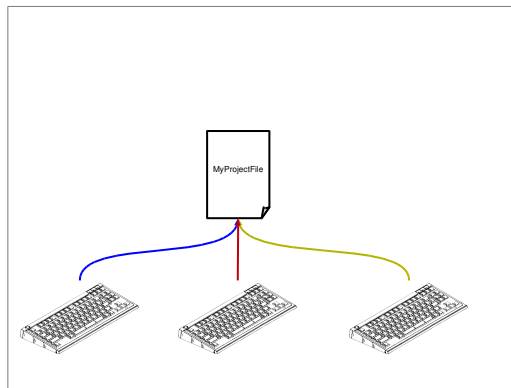


- portabilité
- configuration
  - ▶ temps de compilation vs temps d'exécution
- installation
- système de paquets

Outils : autoconf, dpkg, rpm



# Plan du cours — collaboration



- calcul de modifications entre fichiers source
- application de modifications
- archivage
- gestion des versions
  - ▶ locale vs à distance
  - ▶ centralisée vs distribuée

Outils : diff/patch, git

Services : github, gitorious

# Organisation du cours

- mardi 10h30-13h30, salle 2031  
10h30-11h30 1 heure de cours  
11h30-13h30 2 heures de TP
- 2 groupes de TD (groupe A et groupe B)  
1 groupe chaque semaine
  - ▶ séances groupe A : 05/02, 19/02, 05/03, 19/03, 02/04, 16/04
  - ▶ séances groupe B : 12/02, 26/02, 12/03, 26/03, 09/04, 23/04

## Équipe pédagogique :

- cours : Stefano Zacchioli
- TD (groupe A) : Antoine Durand-Gasselín
- TD (groupe B) : Stefano Zacchioli



Le cours est validé par un examen : en partie écrit et en partie au machine, pour vérifier votre maîtrise des outils de développement dans la pratique.

## Page web du cours

<http://upsilon.cc/zack/teaching/1213/ed6/>

L'inscription à la liste de diffusion des annonces est obligatoire :

- 13ed — annonces

<https://listes.sc.univ-paris-diderot.fr/sympa/info/13ed>

Voir la page du cours pour plus d'information