

Environnements et Outils de Développement

Cours 1 — Introduction

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

Laboratoire PPS, Université Paris Diderot

2013-2014

URL <http://upsilon.cc/zack/teaching/1314/ed6/>
Copyright © 2012-2014 Stefano Zacchioli
License Creative Commons Attribution-ShareAlike 4.0 International License
http://creativecommons.org/licenses/by-sa/4.0/deed.en_US



“Développer” ...

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

“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 :

- 1 implanter une phase d'un **processus de développement logiciel**
- 2 **automatiser** des tâches importantes *et* ennuyeux
- 3 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 :

- 1 implanter une phase d'un **processus de développement logiciel**
 - 2 **automatiser** des tâches importantes *et* ennuyeux
 - 3 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 la version 2.0 de toto sur 1'000 machines ?
 - ▶ combien d'**actions manuelles** dans chaque cas ?

Automatisation des taches

HOW LONG CAN YOU WORK ON MAKING A ROUTINE TASK MORE EFFICIENT BEFORE YOU'RE SPENDING MORE TIME THAN YOU SAVE?
(ACROSS FIVE YEARS)

HOW MUCH TIME YOU SHAVE OFF

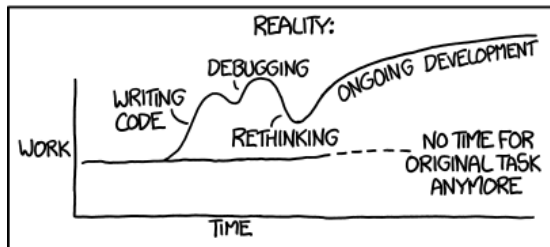
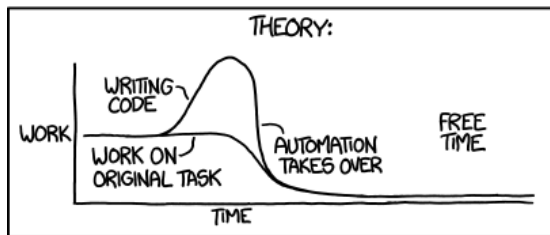
HOW OFTEN YOU DO THE TASK

	50/DAY	5/DAY	DAILY	WEEKLY	MONTHLY	YEARLY
1 SECOND	1 DAY	2 HOURS	30 MINUTES	4 MINUTES	1 MINUTE	5 SECONDS
5 SECONDS	5 DAYS	12 HOURS	2 HOURS	21 MINUTES	5 MINUTES	25 SECONDS
30 SECONDS	4 WEEKS	3 DAYS	12 HOURS	2 HOURS	30 MINUTES	2 MINUTES
1 MINUTE	8 WEEKS	6 DAYS	1 DAY	4 HOURS	1 HOUR	5 MINUTES
5 MINUTES	9 MONTHS	4 WEEKS	6 DAYS	21 HOURS	5 HOURS	25 MINUTES
30 MINUTES		6 MONTHS	5 WEEKS	5 DAYS	1 DAY	2 HOURS
1 HOUR		10 MONTHS	2 MONTHS	10 DAYS	2 DAYS	5 HOURS
6 HOURS				2 MONTHS	2 WEEKS	1 DAY
1 DAY					8 WEEKS	5 DAYS

<http://xkcd.com/1205/>

Automatisation des taches (cont.)

"I SPEND A LOT OF TIME ON THIS TASK.
I SHOULD WRITE A PROGRAM AUTOMATING IT!"



<http://xkcd.com/1319/>

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 développeurs, l'utilisation des outils correspondantes à toute tâche du développement logiciel sera quotidienne. Maîtriser ces outils 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 assomption?)

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

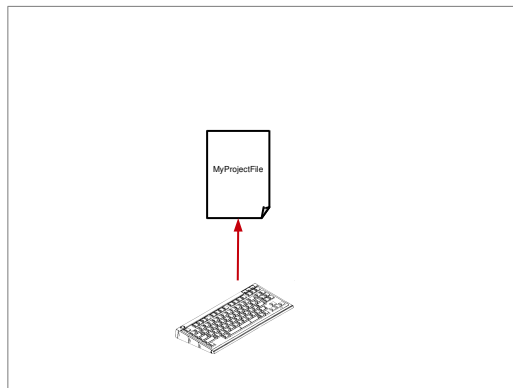
- maîtriser les composantes individuelles, pour mieux comprendre leur interactions et pouvoir en suite maîtriser leur **agrégations**
- le paradigme des **environnements de développement intégré** (IDE) comme orchestrations d'outils plus simple

Méthode du cours (cont.)

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

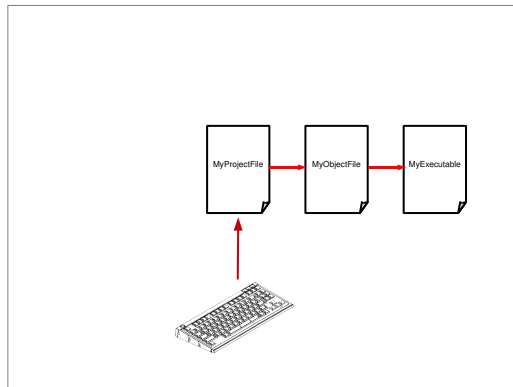
Plan du cours — fichier



- éditeurs de texte ;
spécificités du source code
- efficacité
- compréhension de la syntaxe : indentation, complètement, etc.
- transformation automatisée
- liens entre fichiers source, navigation

Outils : Emacs, ctags, indent, doxygen

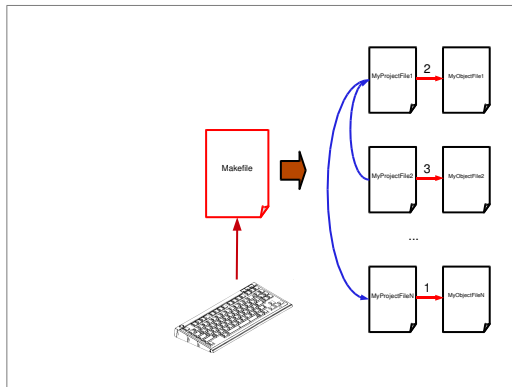
Plan du cours — compilation



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

Outils : gcc, ld

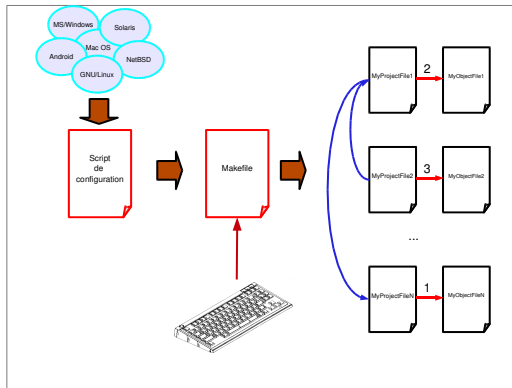
Plan du cours — projet



- calcul des dépendances
- automatisation et minimisation de la chaîne de compilation
- automatisation d'autres tâches
 - ▶ batterie de tests
 - ▶ publication
 - ▶ ...

Outils : make

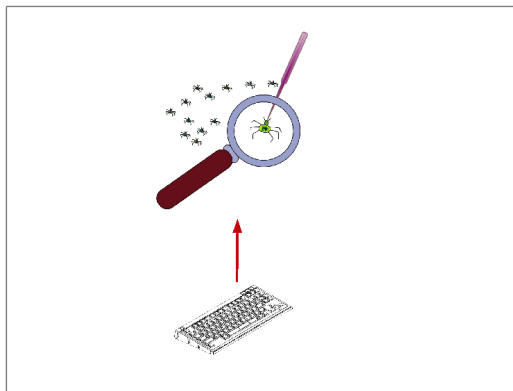
(Plan du cours — configuration)



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

Outils : autoconf, dpkg, rpm

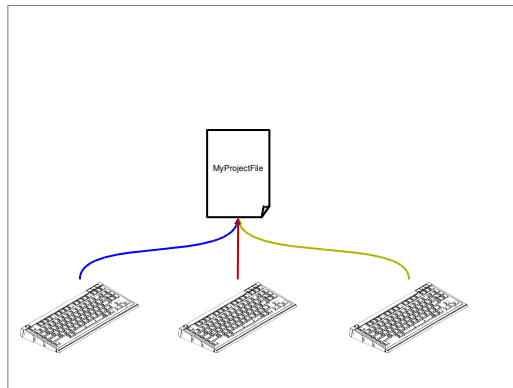
(Plan du cours — exécution)



- reproduire une erreur
- débogage
 - ▶ *stack trace*
 - ▶ exécution pas à pas
- analyse de performances
- analyse de l’empreinte mémoire

Outils : gdb, gprof, valgrind

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, gitlab

Organisation du cours

Équipe pédagogique :

- cours : Stefano Zacchiroli
- TD : Giulio Guerrieri et Antoine Mamcarz
- lundi 13h30-14h30, salle 2031
 - 13h30-14h30 1 heure de cours
 - 14h30-16h30 2 heures de TP
- 2 groupes de TD (groupes : A et B), 1 groupe par semaine

Séances groupe A (avec Giulio Guerrieri)

- ▶ 03/02
- ▶ 17/02
- ▶ 03/03
- ▶ 17/03
- ▶ 31/03

Séances groupe B (avec Antoine Mamcarz)

- ▶ 10/02
- ▶ 24/02
- ▶ 10/03
- ▶ 24/03
- ▶ 07/04

Le cours est validé par :

- **examen** : en partie écrit et en partie au machine, pour vérifier votre maîtrise des outils de développement dans la pratique
- **contrôle continu** : vous devrez rendre chaque TP via DidEL **2 semaines après** la distribution du sujet

Page web

<http://upsilon.cc/zack/teaching/1314/ed6/>

Page DidEL

L'inscription à la page DiDel du cours est **obligatoire** :

<http://didel.script.univ-paris-diderot.fr/claroline/course/index.php?cid=ED6>

Toutes les **annonces** du cours seront envoyées à travers DidEL

Voir la page web du cours pour plus d'information.