

# DevOps

## Déboguer

Thomas Ropars

`thomas.ropars@univ-grenoble-alpes.fr`

2018

# Introduction

Déboguer = trouver et corriger les erreurs dans un programme

- Déboguer des programmes de taille importante peut être complexe
- Un bon programmeur connaît les outils qui peuvent l'aider à déboguer!

# Défi

## Input: des symptômes

- Des tests "fructueux"
  - ▶ Une liste de tests ayant échoués
- Des retours d'utilisateurs
  - ▶ Une liste de problèmes rapportés

# Défi

## Input: des symptômes

- Des tests "fructueux"
  - ▶ Une liste de tests ayant échoués
- Des retours d'utilisateurs
  - ▶ Une liste de problèmes rapportés

## Problèmes

- Le lien entre symptômes et erreurs (malfaçon) n'est pas toujours clair
- Le symptôme peut changer si une autre malfaçon est corrigée
- L'erreur peut être dans la procédure de test
- Certains symptômes sont durs à reproduire
  - ▶ En particuliers ceux liés à des problèmes de *timing*
- Heisenbug

# Quelques conseils

- Supposez toujours que l'erreur vient de vous
- Examinez les résultats intermédiaires
  - ▶ Peut nécessiter d'ajouter du code
  - ▶ Permet de localiser l'erreur
- L'erreur n'est probablement pas où vous pensez
  - ▶ Se méfier des évidences / garder l'esprit ouvert
  - ▶ Revoir sa manière de réfléchir
- Vérifiez vos "inputs"
  - ▶ L'erreur peut être dans la procédure de tests
- Repensez à ce que vous considérez comme acquis
  - ▶ Ce n'est pas parce qu'un code passe des tests qu'il ne contient pas d'erreurs

## Quelques conseils

- Commencez par penser aux problèmes simples
  - ▶ Ordre des paramètres
  - ▶ Initialisation d'une variable
  - ▶ ...
- Demandez de l'aide
  - ▶ Un regard extérieur peut aider
- Vérifiez que vous travaillez sur la bonne version du code
- Faîtes une pause

# Quelques conseils

- Commencez par penser aux problèmes simples
  - ▶ Ordre des paramètres
  - ▶ Initialisation d'une variable
  - ▶ ...
- Demandez de l'aide
  - ▶ Un regard extérieur peut aider
- Vérifiez que vous travaillez sur la bonne version du code
- Faîtes une pause
- Certains outils peuvent vous aider

# Comment comprendre ce qu'il se passe?

## Afficher des messages

- Créer un *log* de l'exécution
- Problèmes:



# Comment comprendre ce qu'il se passe?

## Afficher des messages

- Créer un *log* de l'exécution
- Problèmes:
  - ▶ Les logs peuvent devenir très volumineux
  - ▶ N'aident pas beaucoup pour certains problèmes (problèmes de synchronisation)

# Débogueur

## Principe

Permet d'observer l'évolution d'un programme pendant son exécution

## Exemples

- Gdb (C, C++, Fortran)
- Le débogueur d'Eclipse (Java)
- Valgrind (observer les fuites mémoires)

# Débogueur

## Ce qu'on peut faire avec

- Stopper l'exécution à un point précis (*breakpoint*)
- Exécuter pas à pas
- Observer le contenu des variables
- Changer de contexte d'exécution (d'un thread à l'autre)
- Se déplacer dans la pile d'appels
- Appeler une fonction
- Changer la valeur d'une variable
- ...

# Références

- Notes de S. Easterbrook