

# Systèmes d'exploitation, 2ème année

Fichiers et processus

Yves STADLER

Université Paul Verlaine - Metz

2 janvier 2012

# Agenda

## Plan du cours

- Éditeurs de textes
- Gestion des sources
- Utiliser POSIX sur windows
- Utiliser des versions natives des outils GNU sur windows
- Rédaction de texte avec  $\text{\LaTeX}$

# Éditeurs de textes

## Liste

- Emacs (GUI, debugger, Lisp, metakeys, très personnalisable)
- (g)Vim (Text, léger, modal, home row)
- Pico
- Nano
- Eclipse (Différentes versions selon les langages, mise-à-jour à la volée, debugger)
- Notepad++

## Choix

- Fonctionnalités, adaptation, environnement de travail, choix personnel.

# POSIX sous windows

## Unix avec Cygwin

- Apporte les fonctions POSIX comme fork, signals
- Collection d'outils
- cygwin1.dll : couche API Unix pour windows
- Implémente POSIX en terme d'appel Win32

## Cygwin n'est pas

- Ne permet pas de compiler sous unix et exécuter sur windows
- Ne permet pas de rendre les applications natives windows au courant des fonctions Unix.

# GNU sous windows

## MinGW

- Minimalist GNU for Windows
- Collection d'outils GNU (sed, awk, gcc, ...)
- indépendance vis-à-vis de DLL C propriétaire (dépend cependant de DLL windows standard)

## MinGW n'est pas

- N'apporte pas une implémentation complète de POSIX

# Construction de programmes

- Make
- ANT (multi-plateforme, basé sur XML pour java au départ)
- Autotools, génération des scripts standards utilisés dans les projets GNU (./configure, make)
- Cmake (multi-plateforme, multiples outils : make, xcode, visual studio)

# Debugger/Profile

## Debugueur

- GDB (GNU debugger)
- Insight (GUI GDB)
- DDD (GUI GDB).

## Profileur

- strace (suivi des appels système)
- ltrace (tracer les appels de librairies dynamiques)
- gprof (GNU profiler, analyse le temps passé dans les fonctions)

# Édition de textes avec L<sup>A</sup>T<sub>E</sub>X

## What you see is what you mean

- Distinction totale entre le fond et la forme
- Un fichier .tex décrit le fond (contenu du document) et la structure (je suis un titre de chapitre, je suis un document de type livre, je suis un paragraphe)
- L'entête d'un document L<sup>A</sup>T<sub>E</sub>X permet d'importer les outils de forme (typographie française/anglaise/..., recto ou recto/verso, colonnes, ...)
- Écriture mathématique codifiée et très puissante.

## Analogie avec la programmation

- Le fichier .tex est la source
- Il est compilé avec un compilateur de documents, exemple pdflatex qui génère des documents PDFs.



# Édition de textes avec $\text{\LaTeX}$

## Travailler avec $\text{\LaTeX}$

- Un éditeur de texte
- Une distribution  $\text{\LaTeX}$  (Live-Tex, MikTeX, ...)

## Pourquoi $\text{\LaTeX}$ plutôt que Word

- Word fait ce que  $\text{\LaTeX}$  fait en grande partie
- Word "cache" un certain nombre de concepts de forme (typographie, césures, gestion de styles) que  $\text{\LaTeX}$  oblige à utiliser (comportement par défaut).
- Les comportements par défaut sont là parce que la rédaction d'un document correct suit des règles de style précis.

# Contrôle de sources

## Git

- Système distribué de contrôle et révision de source.
- Ne repose pas sur un serveur central (voir svn)
- Travailler à plusieurs sur de grands projets

## SVN

- Système centralisé de contrôle de sources
- Dérivé de CVS

# Auto-documentation

## Documentation

- Javadoc : documentation pour les projets Java
- Doxygen : documentation pour les projets C++

# Ne pas oublier les outils standards

## Manipulation de texte

- sed, awk
- Expression rationnelle
- Scripts (bash, zsh, ...)

## Beaucoup de sources d'aide avec internet

- Man
- UseNet (serveur de news) en déclin ? ou pas...
- Moteurs de recherches, forums
- Site de l'éditeur
- Stackoverflow (et sites StackExchange)
- Collègues