

A large, semi-transparent watermark of Tux the penguin is centered in the background of the slide.

# **Driver CanPCI, Interface Java**

Christophe Maillot

Xavier Ordoquy

Encadrement par:

Partice Kadionik

A large, semi-transparent watermark of Tux the penguin is centered in the background of the slide.

## **Plan**

- **Introduction**
- **Linux et les drivers**
- **Java**
- **Carte de test**
- **Conclusion**

# Introduction, la carte CanPCI

- **But :**  
test de CAN.
- **Performance :**  
lecture de 24 bits à plus de 50 MHz.
- **Travail à faire :**  
Driver, interface et carte.

## Driver Linux :

- **Généralités,**
- **Les modules,**
- **Drivers et fichiers,**
- **Les IOCTLs,**
- **Le PCI,**
- **Conclusion : le driver canpci.**

# Driver Linux, Généralités

- **Une interface standard :**
  - Des déclarations communes,
  - Une implémentation spécifique au matériel.
- **Les particularités des drivers :**
  - Problèmes de gestion de mémoire,
  - Les entrées/sorties.

# Driver Linux, Les modules

- **Qu'est-ce qu'un module :**
  - Morceau du noyau,
  - Insertion/suppression,
  - Fonctions du driver.
- **Avantages :**
  - Limiter la taille du noyau,
  - Adapter un noyau générique à un système.

# Driver Linux, Drivers et fichiers

- **Le repertoire /dev,**
- **Les majors et minors,**
- **Association fichier/module,**
- **Le nouveau système :**
  - Limitations du système actuel,
  - Le nouveau système : devfs.

# Driver Linux, Drivers et fichiers

- **La lecture/écriture,**
- **L'ouverture/fermeture,**
- **Les opérations diverses,**
- **Les IOCTLs :**
  - Paramétrage du driver
  - Implémentation : une fonction callback

# Driver Linux, le PCI

- **La norme PCI :**
  - Avantages et intérêt par rapport à d'autres bus,
  - Architecture du bus PCI.
- **Utilisation des fonctions du noyau :**
  - Trouver une carte,
  - Dialoguer avec la carte.
- **Nouveautés du noyau 2.4.**

# Driver Linux, conclusion : le driver canpci

- **Reprise des concepts :**
  - Le module Linux,
  - Le fichier /dev/canpci,
  - La communication avec la carte PCI.

# Java :

- **L'interfaçage avec du code natif :**  
**JNI**
- **L'interface graphique :**  
**Swing**

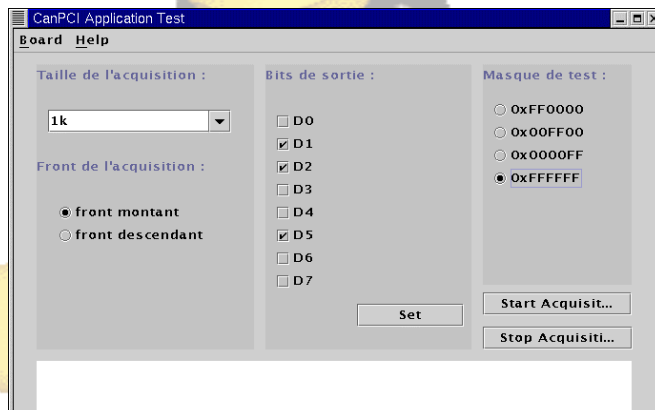
# Java, Java Native Interface

- **Présentation de JNI,**
- **Méthodologie :**
  - Déclarations des méthodes natives,
  - Ecriture du code natif.
- **Possibilités offertes par JNI :**
  - Création d'objets Java,
  - Portabilité du code natif.

# Java, Swing

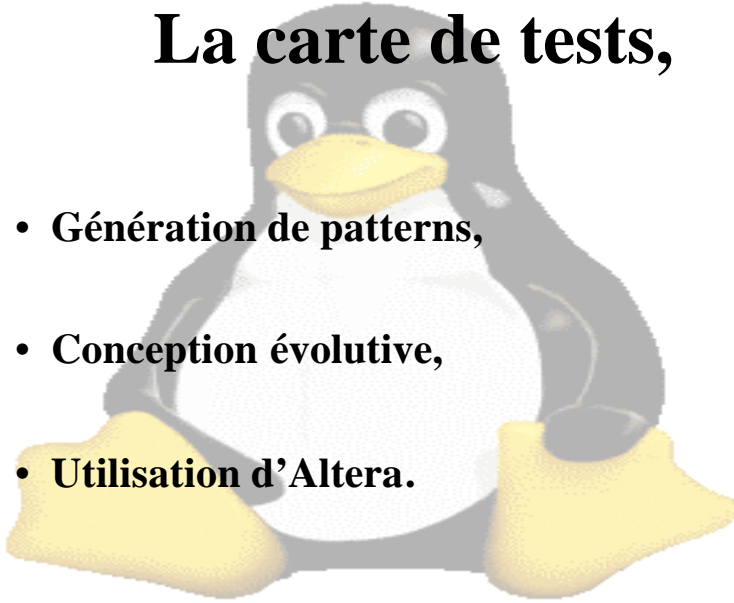
- **Présentation des Swings,**
- **Programmation événementielle,**
- **Dialogue avec la carte CanPCI.**

# Java, Swing



## **La carte de tests,**

- **Génération de patterns,**
- **Conception évolutive,**
- **Utilisation d'Altera.**



## **Conclusion**

