# 130 Mécanismes matériels

## INF3173
## Principes des systèmes d'exploitation

Jean Privat

Université du Québec à Montréal

Hiver 2021

# Mode noyau processeur (privilégié)

## Analogie

- « L'État est une communauté humaine qui, dans les limites d'un territoire, revendique avec succès le monopole de la violence physique légitime. » — Max Weber, *Le Savant et le politique* (1919)
- « Le système d'exploitation est une couche logicielle qui, dans les limites d'un ordinateur, revendique avec succès le monopole du mode noyau. » — Analogie facile...

## Appels système

Permet à un processus de demander un service au SE

- Passage du CPU en mode noyau
- Branchement à une sous-routine spéciale du SE

# Plan

# Interruption matérielle

# Interruption matérielle

## Permettre au matériel de signaler des évènements

- Appui d'une touche, nouveaux paquets réseau, etc.
- Notification d'une commande terminée
- Problème physique
- Etc.

## Mécanisme

- Connexion dédiée: périphériques $\rightarrow$ CPU
  - Pour signaler l'existence d'un évènement
- Le CPU vérifie la présence d'une interruption
  - À chaque instruction
- Si interruption, automatiquement le CPU
  - Sauvegarde des registres (dont le CO)
  - Passe en mode noyau
  - Branche à un endroit spécifique en mémoire

# SE gère les interruptions

## Le noyau au démarrage:

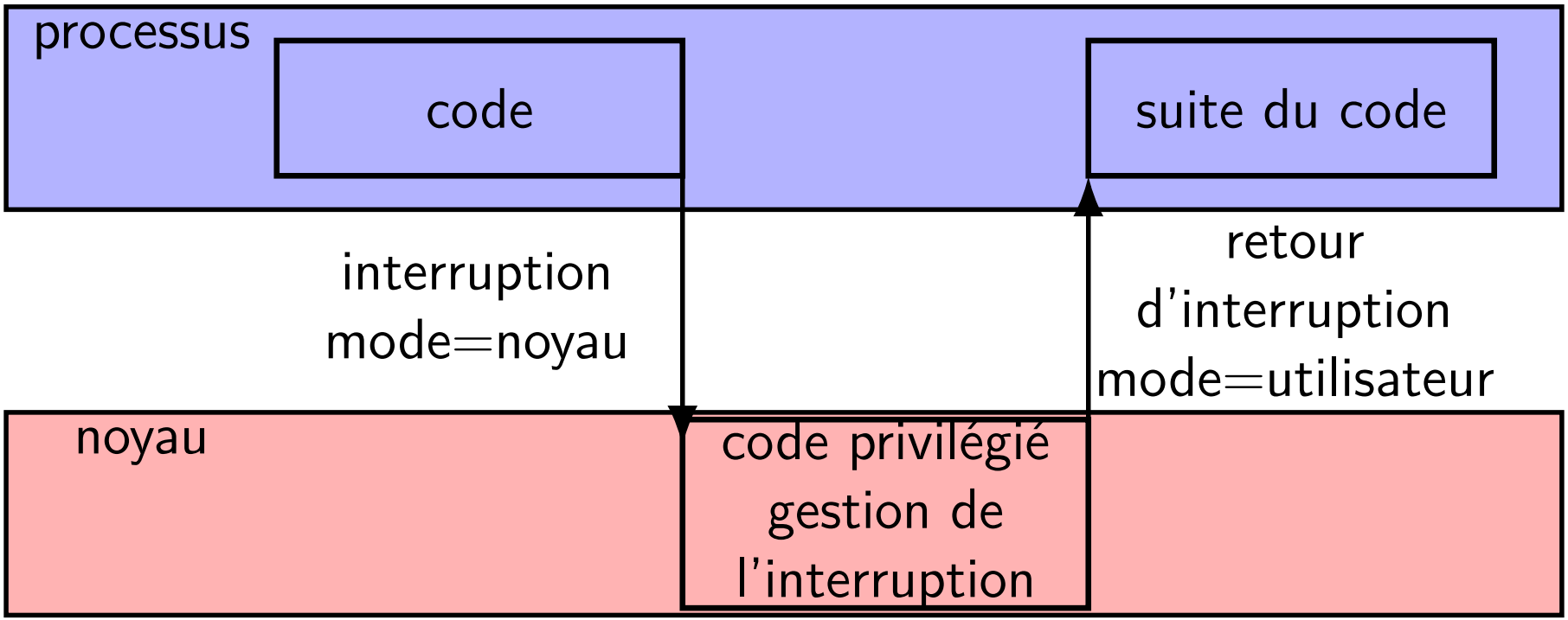
- Configure la machine
- Sous-routines spéciales associées aux interruptions

## Le noyau en cas d'interruption:

- Le processus actif perd le CPU
- Une routine spéciale du noyau est automatiquement invoquée
- Le noyau
  - Sauvegarde les registres
  - Traite efficacement l'interruption
  - Restaure les registres
  - Passe en mode utilisateur

  $\rightarrow$ Le processus s'est rendu compte de rien

# Exemple




- Une interruption matérielle arrive
- Le CPU est donné au noyau qui traite avec le matériel
- Le noyau rend la main au processus

$\rightarrow$ Le processus est interrompu, mais ne s'en rend pas compte

# Interruptions vs. appels système

## Cause

- L'appel système est volontaire
  Le processus fait un appel explicite
- L'interruption est involontaire
  Peut arriver à tout moment

## Mécanismes analogues

- Bascule en mode noyau
- Branchement emplacement dédié du noyau
- Sauvegarde et restauration de registres

## « Interruption logicielle »

- Nom alternatif des appels système
- `int` pour x86, `swi` (*software interrupt*) pour ARM

$\rightarrow$ Cause de la confusion inutile

# Fautes

# Fautes

## Mécanisme: faute CPU

Le CPU lance (lui-même) une interruption matérielle en cas de:

- Instruction inconnue
- Opérandes invalides (division par 0)
- Violation de privilège (mode utilisateur)
- Etc.

On trouve aussi les termes « exception » ou « *trap* »

## Politique

Le système d'exploitation sait

- Gérer les fautes CPU: interruptions matérielles classiques
- Déterminer le responsable: le processus qui a été interrompu

# Exemple de scénario

- Un processus exécute une instruction privilégiée
- Le CPU refuse (mode utilisateur) et génère une faute
- Le SE s'exécute alors:
  - Inspecte les registres et la mémoire
  - Détermine le processus coupable
  - Lui envoie un signal (`kill`)
  - Ce qui termine le processus

## Justice implacable

SE = investigue, arrête, condamne et exécute les processus délinquants

## Exemple: division par zéro

```
int main(int argc, char *argv[]) { return 0/0; }
```

# Horloge programmable

# Horloge programmable

## Problèmes

Comment attendre des échéances ?

- Faire une pause quelques secondes
- Gérer les expirations (timeouts)

Comment récupérer un CPU accaparé par un processus ?

- Calcul intensif
- Boucle infinie

## Mécanisme

- Un matériel spécial

$\rightarrow$ une composante dédiée sur la carte mère

$\rightarrow$ ou directement le contrôleur d'interruption

- Décrémente un compteur
- Lève une interruption quand il atteint 0

# Exemple de politique

- Le SE programme l'horloge
- Puis il donne la main à un processus
- Le processus bloque le CPU dans une boucle infinie
- Le délai programmé de l'horloge expire
- Une interruption est levée
- Le CPU est rendu au système

**Question.** C'est le processus ou le processeur qui est en boucle infinie ?

## Multitâche

- C'est la base du multitâche préemptif
- Permet de répartir le CPU entre plusieurs processus

$\rightarrow$ On y reviendra

# 3 types d'horloges dans un ordinateur

## Horloge programmable

- Pour lever des interruptions
- Analogie: minuterie

## Signal d'horloge

- Rythme le fonctionnement électronique (CPU, RAM, Bus, etc.)
- Analogie: métronome

## Horloge temps réel

- Maintient la date et l'heure réelle
- Alimentation autonome avec une pile
- Analogie: horloge murale

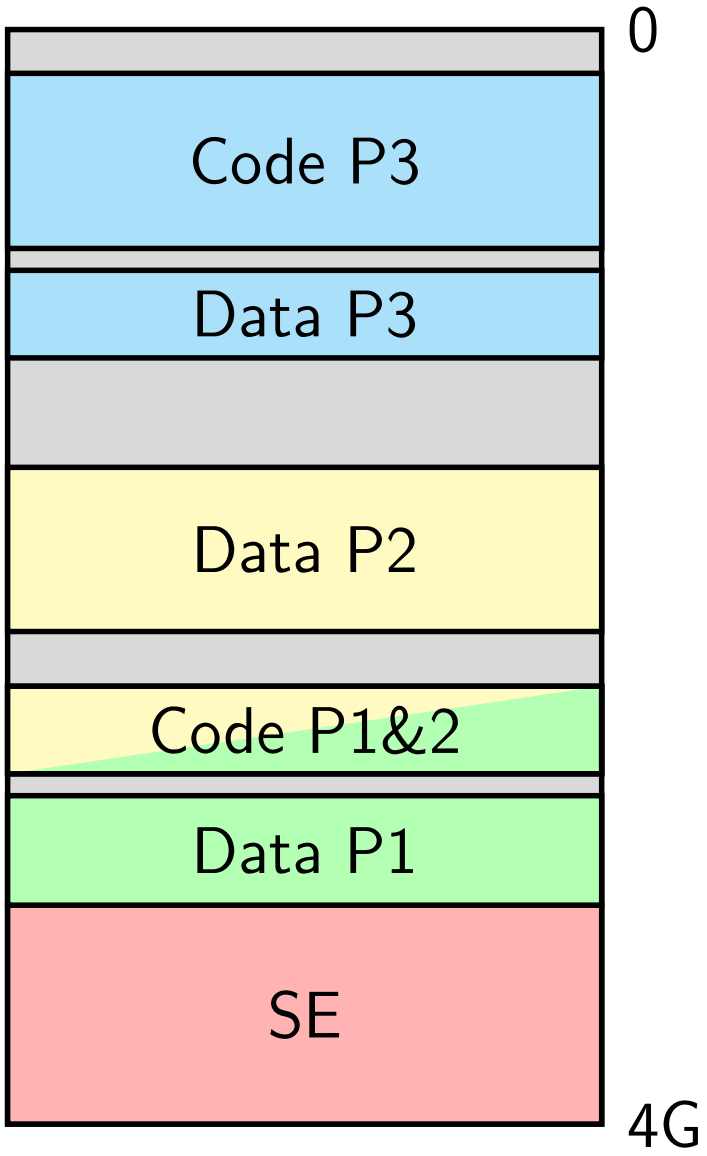
# Protection mémoire

# Organisation mémoire d'un programme

## En mémoire, il y a

- Code du programme (en langage machine)
- Données (statiques, pile, tas, etc.)
- Bibliothèques

# Défi: plusieurs programmes à la fois




## Chaque processus

- A accès (lecture/écriture) qu'à son propre espace
- Tout accès en dehors est physiquement interdit

## Le SE

- A accès à toute la mémoire
- Gère les limites physiques des programmes

# Mécanisme

- On peut marquer des zones mémoires comme valides ou invalides
- Le mode noyau est nécessaire pour changer les zones
- Le CPU peut **efficacement** déterminer la validité d'une adresse

$\rightarrow$ Plusieurs approches possibles, on y reviendra

- Un accès en dehors d'une zone valide lève une faute

$\rightarrow$ le CPU vérifie chaque accès fait à la mémoire

# Politique

- Le système rend valide les zones mémoire d'un processus
- Puis il lui donne la main
- Le processus accède dans ses zones
  - Tout va bien
- Le processus accède en dehors
  - Le CPU lève une faute
  - Le SE prend la main
  - Envoie un signal au processus fautif (`kill`)
  - Ce qui le termine

```
#include <stdlib.h>
int main(int argc, char *argv[]) {
  int *i = NULL; return *i;
}
```

Nom habituel: « faute de segmentation » (*segmentation fault*)

# Mémoire virtuelle et pagination

De nos jours quasiment

- tout processeur
- et système d'exploitation

Utilisent

- De la mémoire virtuelle
- Plus précisément de la pagination
- Avec des modes de protection

On y reviendra plus tard

# Conclusion : Le SE au centre