TSE2531: Technique des Systèmes d'Exploitation

Interruptions, déroutements et appels système

Par: Miguel Landry FOKO SINDJOUNG, PhD

Institut Universitaire de Technologie Fotso Victor de Bandjoun
Filière: Génie-Informatique
Niveau: DUT 2

Année académique 2022-2023

Objectifs

- Comprendre le mode d'exécution des programmes sur une machine
- Définir et comprendre le mécanisme des interruptions
- Donner le rôle des interruptions
- Savoir à quoi renvoient les notions d'appel système et de déroutement

Exécution de programmes (1)

Une machine est composée schématiquement :

- D'un processeur (aussi appelée la C.P.U.)
- D'une mémoire principale
- D'organes d'E/S

Exécution de programmes (1)

Une machine est composée schématiquement :

- D'un processeur (aussi appelée la C.P.U.)
- D'une mémoire principale
- D'organes d'E/S

L'ensemble des registres spécialisés forment le **mot d'état du processeur** (M.E.P. ou P.S.W. pour Processor Status Word)

Exécution de programmes (2)

Parmi les registres du processeur, on retrouve :

- Le compteur ordinal (CO) qui contient l'adresse de la prochaine instruction à exécuter ;
- Le mode d'exécution (MODE) qui peut être maître ou esclave ;
- Le masque d'interruptions que nous détaillerons plus tard.
- Un ou plusieurs pointeur(s) de pile.
- etc.

Exécution de programmes (2)

Parmi les registres du processeur, on retrouve :

- Le compteur ordinal (CO) qui contient l'adresse de la prochaine instruction à exécuter ;
- Le mode d'exécution (MODE) qui peut être maître ou esclave ;
- Le masque d'interruptions que nous détaillerons plus tard.
- Un ou plusieurs pointeur(s) de pile.
- etc.

La notion de mode a été introduite essentiellement pour des raisons de protection, afin qu'un programme quelconque ne puisse accéder à des zones ou à des registres propres au S.E.

Exécution de programmes (3)

- Dans la pratique, la CPU distingue les instructions normales et les instructions privilégiées
- Ces dernières ne sont utilisables qu'en mode maître
- Elles permettent en général la modification des registres spécialisés et le dialogue avec les unités d'E/S
- Une **exécution** est une évolution discrète de l'état de la machine
- Cet état est donné par le contenu de la mémoire et la valeur des registres de la CPU

Le mécanisme des interruptions (1)

Dans tous les types de système, il est toujours nécessaire de considérer un **travail courant** (le programme en cours d'exécution) et un **travail exceptionnel** dont le but est de traiter un événement :

- Dans les systèmes de conduite de processus, certains événements importants (voir même graves) doivent être pris en compte dans les délais les plus brefs

Le mécanisme des interruptions (1)

Dans tous les types de système, il est toujours nécessaire de considérer un **travail courant** (le programme en cours d'exécution) et un **travail exceptionnel** dont le but est de traiter un événement :

- Dans les systèmes de conduite de processus, certains événements importants (voir même graves) doivent être pris en compte dans les délais les plus brefs
    - En d'autres termes, il faut donc interrompre le travail courant (relevés des capteurs), pour exécuter un programme prioritaire.
- Il existe toujours un dialogue entre l'U.C. et les organes d'E/S

Le mécanisme des interruptions (1)

Dans tous les types de système, il est toujours nécessaire de considérer un **travail courant** (le programme en cours d'exécution) et un **travail exceptionnel** dont le but est de traiter un événement :

- Dans les systèmes de conduite de processus, certains événements importants (voir même graves) doivent être pris en compte dans les délais les plus brefs
    - En d'autres termes, il faut donc interrompre le travail courant (relevés des capteurs), pour exécuter un programme prioritaire.
- Il existe toujours un dialogue entre l'U.C. et les organes d'E/S

Les interruptions permettent d'interrompre provisoirement le déroulement d'un programme en cours pour exécuter une routine considérée comme prioritaire

Le mécanisme des interruptions (2)

En cas d'interruption :

- On associe à chaque cause d'interruption un numéro k qui l'identifie

Le mécanisme des interruptions (2)

En cas d'interruption :

- On associe à chaque cause d'interruption un numéro k qui l'identifie
- On dispose également dans les adresses basses de la mémoire d'une table appelée le vecteur d'interruptions (vi)
- Les cases vi[k] de cette table contiennent l'adresse de la routine à exécuter lors d'une interruption de cause k
- Cette routine est appelée le traitant d'interruption de cause k.

Le mécanisme des interruptions (3)

Plus précisément, lors d'une interruption de cause k, la CPU effectue dès la fin de l'instruction en cours les actions suivantes :

1. Sauvegarder la valeur du compteur ordinal et le mode d'exécution (dans une pile ou dans une case mémoire particulière suivant les C.P.U.) ;

Le mécanisme des interruptions (3)

Plus précisément, lors d'une interruption de cause k, la CPU effectue dès la fin de l'instruction en cours les actions suivantes :

1. Sauvegarder la valeur du compteur ordinal et le mode d'exécution (dans une pile ou dans une case mémoire particulière suivant les C.P.U.) ;
2. Passer en mode maître ;

Le mécanisme des interruptions (3)

Plus précisément, lors d'une interruption de cause k, la CPU effectue dès la fin de l'instruction en cours les actions suivantes :

1. Sauvegarder la valeur du compteur ordinal et le mode d'exécution (dans une pile ou dans une case mémoire particulière suivant les C.P.U.) ;
2. Passer en mode maître ;
3. Forcer dans le compteur ordinal la valeur vi[k], c'est à dire l'adresse de la première instruction de la routine associée à l'interruption de cause k.

Le mécanisme des interruptions (3)

Plus précisément, lors d'une interruption de cause k, la CPU effectue dès la fin de l'instruction en cours les actions suivantes :

1. Sauvegarder la valeur du compteur ordinal et le mode d'exécution (dans une pile ou dans une case mémoire particulière suivant les C.P.U.) ;
2. Passer en mode maître ;
3. Forcer dans le compteur ordinal la valeur vi[k], c'est à dire l'adresse de la première instruction de la routine associée à l'interruption de cause k.

L'interruption est donc un mécanisme matériel puisque la sauvegarde et l'initialisation du compteur ordinal à partir du vecteur d'interruptions sont des opérations réalisées par la CPU

Le mécanisme des interruptions (4)

Le traitant représente la partie logicielle du mécanisme d'interruption. Sa structure est la suivante :

1. Sauvegarder la valeur des registres de la CPU (dans un emplacement particulier de la mémoire). Cette étape est couramment appelée la sauvegarde du contexte

Le mécanisme des interruptions (4)

Le traitant représente la partie logicielle du mécanisme d'interruption. Sa structure est la suivante :

1. Sauvegarder la valeur des registres de la CPU (dans un emplacement particulier de la mémoire). Cette étape est couramment appelée la sauvegarde du contexte

2. Traiter la cause de l'interruption.

Le mécanisme des interruptions (4)

Le traitant représente la partie logicielle du mécanisme d'interruption. Sa structure est la suivante :

1. Sauvegarder la valeur des registres de la CPU (dans un emplacement particulier de la mémoire). Cette étape est couramment appelée la sauvegarde du contexte
2. Traiter la cause de l'interruption.
3. Restaurer la valeur des registres de la CPU et le mode du programme interrompu. C'est la restauration du contexte.

Le mécanisme des interruptions (4)

Le traitant représente la partie logicielle du mécanisme d'interruption. Sa structure est la suivante :

1. Sauvegarder la valeur des registres de la CPU (dans un emplacement particulier de la mémoire). Cette étape est couramment appelée la sauvegarde du contexte
2. Traiter la cause de l'interruption.
3. Restaurer la valeur des registres de la CPU et le mode du programme interrompu. C'est la restauration du contexte.
4. Forcer dans le compteur ordinal la valeur préalablement sauvegardée.

Le mécanisme des interruptions (5)

Deux conclusions à partir d'ici :

1. les traitants d'interruption s'exécutent en mode maître (donc avec des droits étendus)

Le mécanisme des interruptions (5)

Deux conclusions à partir d'ici :

1. les traitants d'interruption s'exécutent en mode maître (donc avec des droits étendus)
2. l'exécution du programme interrompu n'est pas perturbée par le traitement de l'interruption

Le mécanisme des interruptions (6)

Les principales utilisations du processus d'interruption sont les suivantes : (1)

- Interruption logicielle (ou déroutement) provoquée par la CPU lors de la détection d'une situation anormale. Par exemple :
  - Appel explicite du S.E.
  - Instruction incorrecte ou inconnue
  - Violation de privilège
  - Dépassement de capacité
  - Division par zéro
  - Tentative d'accès à une zone protégée de la mémoire.

Le mécanisme des interruptions (7)

Les principales utilisations du processus d'interruption sont les suivantes : (2)

- Interruption matérielle générée par une unité externe à la CPU afin de lui signaler l'apparition d'un événement extérieur. Par exemple :
  - Fin d'une E/S
  - Impulsion d'horloge
  - Changement d'état d'un des périphériques
  - Présence d'une valeur intéressante sur un capteur.

Les interruptions matérielles (1)

- Les interruptions matérielles se présentent comme un ensemble de fils numérotés reliant la CPU et les circuits externes de la machine

Les interruptions matérielles (1)

- Les interruptions matérielles se présentent comme un ensemble de fils numérotés reliant la CPU et les circuits externes de la machine
- La présence d'un signal sur un de ces fils provoque une interruption du programme en cours d'exécution de fils numérotés reliant la CPU et les circuits externes de la machine
- Le numéro de cette interruption est directement lié au fils qui l'a déclenchée

En résumé, un circuit extérieur génère une interruption sur la CPU afin de lui signaler un événement. Cette interruption stoppe le programme en cours pour lancer une routine du S.E. L'exécution de cette routine permet, dans les meilleurs délais, la prise en compte par le S.E. de l'événement extérieur.

Les interruptions matérielles (2)

Le système hiérarchisé d'interruptions :

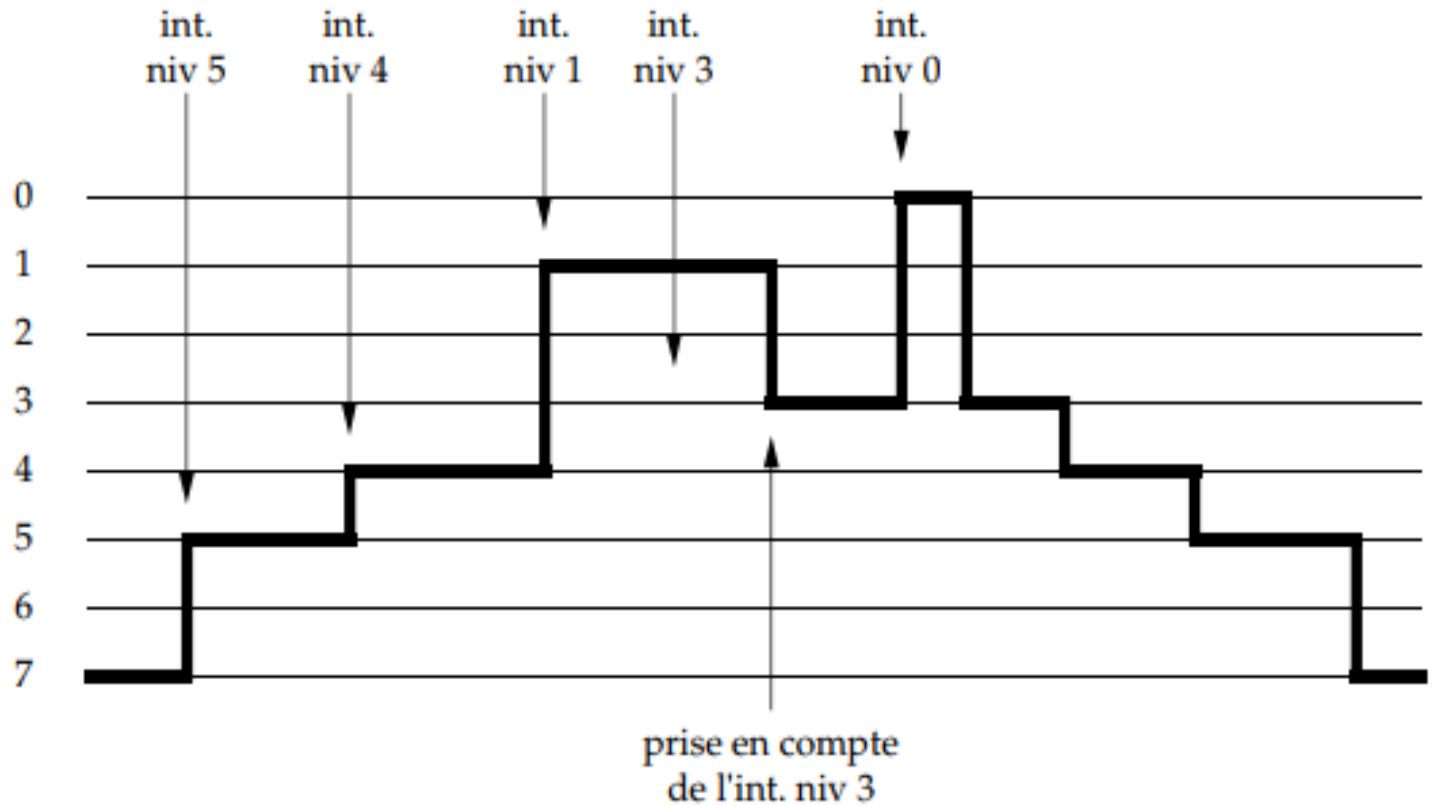

Figure – Effets de la hiérarchisation d'un système d'interruption

Les interruptions matérielles (3)

Commande du système d'interruption : (1)

- État désarmé : le niveau n'accepte aucune demande d'interruption.

Les interruptions matérielles (3)

Commande du système d'interruption : (1)

- État désarmé : le niveau n'accepte aucune demande d'interruption.
- État armé : le niveau accepte et mémorise une demande d'interruption. On peut armer ou désarmer un niveau d'interruption par programme en utilisant des instructions privilégiées. Cette possibilité est donc réservée au S.E.

Les interruptions matérielles (3)

Commande du système d'interruption : (1)

- État désarmé : le niveau n'accepte aucune demande d'interruption.
- État armé : le niveau accepte et mémorise une demande d'interruption. On peut armer ou désarmer un niveau d'interruption par programme en utilisant des instructions privilégiées. Cette possibilité est donc réservée au S.E.
- État masqué : le niveau a été inhibé par programme de sorte que l'interruption a pu être mémorisée mais ne peut être prise en compte par la CPU.

Les interruptions matérielles (4)

Commande du système d'interruption : (2)

- État d'attente : l'interruption peut être prise en compte immédiatement si deux conditions sont remplies :
  - Aucun niveau de priorité supérieure n'est en état d'attente ;
  - La CPU se trouve dans une phase interruptible (fin d'instruction). Le niveau passe alors à l'état actif.

Les interruptions matérielles (4)

Commande du système d'interruption : (2)

- État d'attente : l'interruption peut être prise en compte immédiatement si deux conditions sont remplies :
  - Aucun niveau de priorité supérieure n'est en état d'attente ;
  - La CPU se trouve dans une phase interruptible (fin d'instruction). Le niveau passe alors à l'état actif.
- État actif : il implique la prise en compte de l'interruption par la CPU et dure pendant toute la durée du traitant d'interruption.

Les appels systèmes (1)

- Nous avons vu précédemment que les programmes utilisateur s'exécutent en mode esclave

Les appels systèmes (1)

- Nous avons vu précédemment que les programmes utilisateur s'exécutent en mode esclave
- Les instructions privilégiées permettant la programmation des E/S leur sont donc interdites

Les appels systèmes (1)

- Nous avons vu précédemment que les programmes utilisateur s'exécutent en mode esclave
- Les instructions privilégiées permettant la programmation des E/S leur sont donc interdites
- Dans ces conditions, toute demande d'E/S et plus généralement toutes les actions demandant des droits étendus, passent par une requête en bonne et due forme au S.E

Les appels systèmes (1)

- Nous avons vu précédemment que les programmes utilisateur s'exécutent en mode esclave
- Les instructions privilégiées permettant la programmation des E/S leur sont donc interdites
- Dans ces conditions, toute demande d'E/S et plus généralement toutes les actions demandant des droits étendus, passent par une requête en bonne et due forme au S.E
- Cette requête est réalisée par le truchement d'une instruction de la CPU qui provoque une interruption : le **SVC** pour **SuperVisor Call** ou encore **TRAP**

Les appels systèmes (2)

La solution précédente a les avantages suivants :

- L'interruption provoque un branchement vers le traitant d'interruption mais aussi un changement de mode : il y a donc un passage automatique du programme utilisateur en mode esclave au S.E. en mode maître.

Les appels systèmes (2)

La solution précédente a les avantages suivants :

- L'interruption provoque un branchement vers le traitant d'interruption mais aussi un changement de mode : il y a donc un passage automatique du programme utilisateur en mode esclave au S.E. en mode maître.

- Il existe un et un seul point d'entrée vers le S.E. pour les processus utilisateur : il est donc plus facile (du point de vue du concepteur du système) de sécuriser l'appel des primitives système.

Les appels systèmes (2)

La solution précédente a les avantages suivants :

- L'interruption provoque un branchement vers le traitant d'interruption mais aussi un changement de mode : il y a donc un passage automatique du programme utilisateur en mode esclave au S.E. en mode maître.
- Il existe un et un seul point d'entrée vers le S.E. pour les processus utilisateur : il est donc plus facile (du point de vue du concepteur du système) de sécuriser l'appel des primitives système.
- Si on part du principe que le vecteur d'interruptions se trouve dans une zone inaccessible au programme utilisateur, alors ce dernier n'a aucun moyen de passer en mode maître et l'instruction SVC est le seul point de passage.

Les déroutements (1)

Un déroutement est une interruption qui intervient lorsqu'une anomalie a été détectée dans le déroulement d'une instruction, empêchant ainsi son exécution. On distingue trois types de causes :

1. Données incorrectes (division par zéro, débordement arithmétique, etc.) ;

Les déroutements (1)

Un déroutement est une interruption qui intervient lorsqu'une anomalie a été détectée dans le déroulement d'une instruction, empêchant ainsi son exécution. On distingue trois types de causes :

1. Données incorrectes (division par zéro, débordement arithmétique, etc.) ;
2. Tentative de violation d'une protection et/ou d'une interdiction (violation de protection mémoire, utilisation d'une instruction privilégiée en mode esclave, etc.) ;

Les déroutements (1)

Un déroutement est une interruption qui intervient lorsqu'une anomalie a été détectée dans le déroulement d'une instruction, empêchant ainsi son exécution. On distingue trois types de causes :

1. Données incorrectes (division par zéro, débordement arithmétique, etc.) ;
2. Tentative de violation d'une protection et/ou d'une interdiction (violation de protection mémoire, utilisation d'une instruction privilégiée en mode esclave, etc.) ;
3. Impossibilité d'exécution d'une instruction (instruction inconnue ou instruction optionnelle absente de la configuration utilisée, etc.).

Les déroutements (2)

Un déroutement est une interruption qui intervient lorsqu'une anomalie a été détectée dans le déroulement d'une instruction, empêchant ainsi son exécution. On distingue trois types de causes :

- Selon la cause d'un déroutement, on peut éventuellement en supprimer l'effet : Par exemple, on peut récupérer les erreurs arithmétiques ou encore les lectures au delà de la fin de la mémoire

Les déroutements (2)

Un déroutement est une interruption qui intervient lorsqu'une anomalie a été détectée dans le déroulement d'une instruction, empêchant ainsi son exécution. On distingue trois types de causes :

- Selon la cause d'un déroutement, on peut éventuellement en supprimer l'effet : Par exemple, on peut récupérer les erreurs arithmétiques ou encore les lectures au delà de la fin de la mémoire
- En résumé, le mode esclave, les déroutements vers le S.E. en cas d'erreur et le mécanisme des appels système imposent un cadre strict pour l'exécution des programmes utilisateur.

Les déroutements (2)

Un déroutement est une interruption qui intervient lorsqu'une anomalie a été détectée dans le déroulement d'une instruction, empêchant ainsi son exécution. On distingue trois types de causes :

- Selon la cause d'un déroutement, on peut éventuellement en supprimer l'effet : Par exemple, on peut récupérer les erreurs arithmétiques ou encore les lectures au delà de la fin de la mémoire
- En résumé, le mode esclave, les déroutements vers le S.E. en cas d'erreur et le mécanisme des appels système imposent un cadre strict pour l'exécution des programmes utilisateur.
- Les systèmes d'exploitation récents sont dits dirigés par les interruptions car ils ne s'exécutent que sur demande explicite