

EXÉCUTION DE PROGRAMMES (1)

■ Principe

- Programme chargé en MC
 - Devient processus
- Exécution par le processeur
 - Exécution séquentielle
 - Un seul processus
 - Une seule instruction par processus

EXÉCUTION DE PROGRAMMES (2)

- ❑ **2 Types de programmes :**
 - Programmes du système
 - ❑ Programmes sûrs (sans erreurs)
 - ❑ Programmes spécifiques
 - Gestion et contrôle de l'utilisation d'une machine
 - Programmes utilisateurs
 - ❑ Exploitent une machine
 - ❑ Programmes non fiables
 - ❑ Contrôler leur exécution

EXÉCUTION DE PROGRAMMES (3)

❑ Modes d'exécution des processus

○ Mode Système

❑ Privilégié, Maître, Master, Superviseur, Système

❑ Réservé aux programmes du système

○ Accès à toute la mémoire

○ Exécution de toutes les instructions du processeur

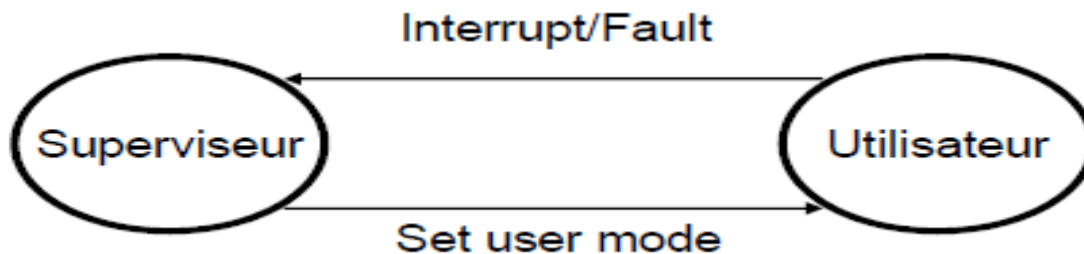
○ Jeu d'instructions complet

EXÉCUTION DE PROGRAMMES (4)

- ❑ Mode Programme
 - ❑ Non privilégié, Esclave, Slave, Utilisateur, User, Programme
- Mode des programmes utilisateurs
- Restrictions
 - ❑ Accès mémoire limité (limites du programme)
 - ❑ Jeu d'instructions réduit
 - Ne peut pas exécuter tout le jeu d'instructions du processeur

EXÉCUTION DE PROGRAMMES (5)

- Possibilité de passage du mode utilisateur au mode système
 - À travers les appels système



APPELS SYSTÈME (1)

Programme utilisateur

- Propres fonctions
 - Calcul d'une fonction mathématique
 - Tri d'un ensemble de données
- Fonctions du système
 - Non exécutables directement
 - Utilisables à travers des appels système
 - Exemples : Fonctions d'E/S, instructions manipulant des protections mémoires, ...

APPELS SYSTÈME (2)

- ❑ **Utilisation dans les langages de programmation**
- Appel par interruption
 - ❑ Par exemple, instruction INT de l'Assembleur
- Inclusion de bibliothèques ou de packages
 - ❑ C, C++, Java

APPELS SYSTÈME (2)

Principe

- Instruction d'appel système
 - Exécutée par un programme user
 - Provoque une INTERRUPTION
 - Passage au mode Maître
 - Exécution du service demandé
 - Retour au programme appelant
 - Suite de l'exécution en mode programme

CONTEXTE D'UN PROCESSUS (1)

- Dans les systèmes multi-tâches ou multi-utilisateurs, plusieurs processus cohabitent et se partagent la mémoire centrale et le processeur central
- il est possible qu'un processus soit **interrompu** (par un autre processus par exemple)

CONTEXTE D'UN PROCESSUS (2)

- Un processus P en exécution peut être interrompu par :
 - une interruption involontaire (par un autre processus par exemple)
 - Une interruption volontaire : correspond à une demande d' une ressource :
 - demande d'ouverture d'un fichier,
 - demande d'impression,
 - demande de lecture du clavier,
 - émission et/ou réception d'un message via la carte réseau
 - etc.

CONTEXTE D'UN PROCESSUS (3)

- le système interrompt momentanément l'exécution du processus P (en un point interruptible)
- Lance le processus système correspondant au traitement de l'interruption
- L'exécution du processus P sera reprise ultérieurement.
- Il est donc nécessaire de sauvegarder certaines informations indispensables à cette reprise (contenu des registres du processeur central au moment de l'interruption)
- Cet ensemble d'information est appelé "*contexte*" du processus

SYSTÈMES D'INTERRUPTIONS (1)- PRÉSENTATION

Mécanisme de base

- Signal technologique asynchrone (physique)
- **Arrêter** le programme en cours d'exécution pour **exécuter** un autre programme (programme de l'OS)
 - Commutation de contexte
- L'arrêt ne peut se faire qu'entre deux instructions
- Cause
 - Interne (**Déroutement**) ou externe (**Interruption**) **par rapport** au programme en cours d'exécution

SYSTÈMES D'INTERRUPTIONS (2)- INTERRUPTION

- Les interruptions sont dues aux périphériques ou à des événements extérieurs
- La gestion des entrées-sorties nécessite le plus grand nombre d'interruptions
- Exemple :
 - Mise en service d'un périphérique
 - Contrôleur de disque signale que la tâche préalablement demandée a été accomplie
 - Imprimante prête à recevoir un document

SYSTÈMES D'INTERRUPTIONS (2)- DÉROUTEMENT

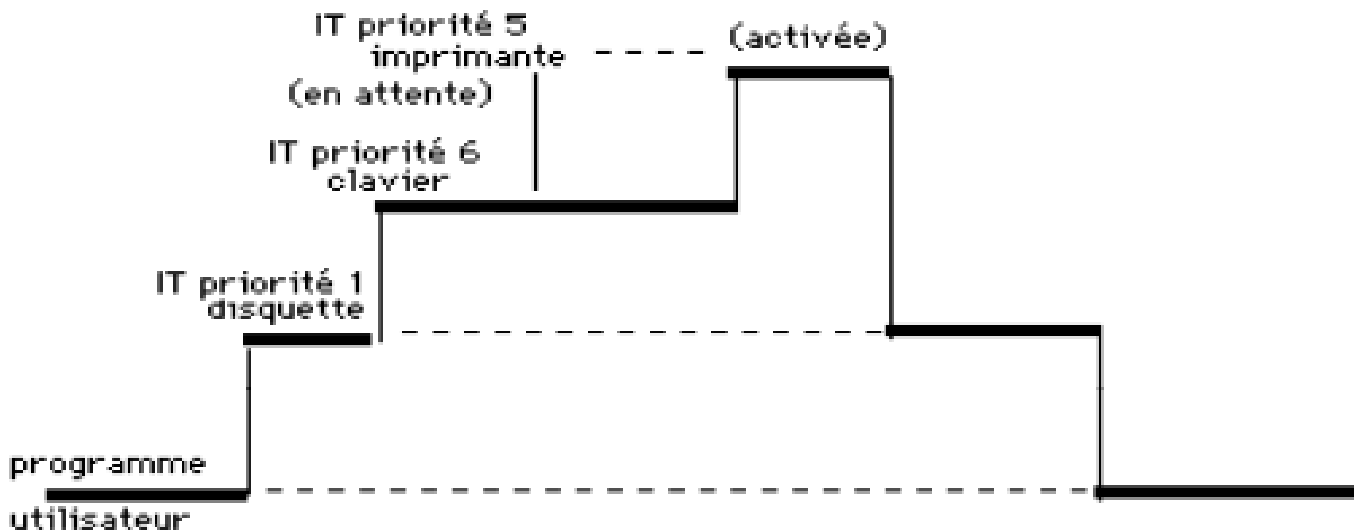
- Les déroutements sont des « interruptions » interne au programme en cours d'exécution.
- Elles peuvent être de deux sortes :
 - Les interruptions dues aux erreurs détectées lors de l'exécution d'un processus
 - dépassement de capacité,
 - erreur d'adressage,
 - accès à une zone mémoire protégée,
 - division par zéro, etc.
 - Les interruptions logicielles qui permettent aux programmes utilisateurs de faire des appels systèmes (CVS : Call SuperVisor en anglais):
 - pour signaler leur terminaisons,
 - pour demander l'allocation d'une ressource partagée, etc.

INTERRUPTIONS (1)-MÉCANISME GÉNÉRAL

- Lorsqu'une interruption survient, le processeur achève l'exécution de l'instruction en cours, puis il se produit :
- Sauvegarde du contexte du processus interrompu
- Chargement du contexte du programme d'interruption (contexte actif) et passage en mode système (ou superviseur)
- Exécution du programme d'interruption
- Retour au programme interrompu en restaurant le contexte et en repassant en mode utilisateur.

INTERRUPTIONS (2)-PRIORITÉS

- Le système d'interruption est en général hiérarchisé.
 - A chaque signal d'IT est affecté un niveau de priorité.
- Si un conflit se présente, l'IT de plus forte priorité sera traitée la première



INTERRUPTIONS (3)- MASQUAGE DES INTERRUPTIONS

- Certaines interruptions sont tellement d'importance qu'il ne doit pas être possible d'interrompre leur traitement.
- On **masquera** alors les autres interruptions pour empêcher leur prise en compte.
- Certaines interruptions sont **non-masquables** : on les prend obligatoirement en compte.

INTERRUPTIONS (4)- MASQUAGE DES INTERRUPTIONS

- Une interruption masquée n'est pas ignorée : elle est prise en compte dès qu'elle est démasquée
- Au contraire, une interruption peut-être **désarmée** : elle sera ignorée. Par défaut, les interruptions sont évidemment **armées**