# Programme du cours de Programmation système

* Rappels systèmes
* Techniques de programmation système
* Technologies systèmes émergentes.

# Introduction

# Rappels systèmes

## Cycle de vie d’un programme de point de vue du système

### Programmation

Problème -> Algorithme -> Langage de programmation -> Programme sur papier (Avant) -> Editeur de texte -> Programme source.

Programme source : est la transcription d’un algorithmique en un programme en utilisant un langage de programmation.

Après le programme source on a le pré-compilateur pour gérer les constantes et l’importation d’un code source réutilisable. Il produit un fichier qui n’est rien d’autre que le programme source avec des enrichissements.

### La phase de vérification du code

Compilateur -> programme assembleur -> Assembleur -> Programme objet + bibliothèques -> Editeur de lien -> Programme exécutable

L’assembleur génère : un descripteur de programme (Il contient les informations nécessaires pour l’exécution du programme), table de symbole (indique les symboles utilisés non définies), le code exécutable.

Forme générale des inscriptions : Code de référence, opérande 1 et opérande 2.

L’Editeur de lien génère : Un descripteur (après avoir résolu les liens), Une table de symbole et un code exécutable.

### Exécution

Le fichier exécutable est pris en charge par le système de gestion de fichiers.

* + Interpréteur de commande récupère la commande entrée ou l’icône cliquée
  + Initialiseur créer le contexte d’exécution du programme en créant le descripteur de processus (code exécutable, Pile d’exécution et la zone des variables. Après le chargement du code par le chargeur, l’initialiseur place le descripteur dans la file d’ordonnancement

Remarque : Un fichier est un descripteur (les métadonnées d’un fichier), la structure de données de stockage et les données.

## Partage de l’unité central

Il se repose sur deux concepts de base : Entité exécutable, le contexte de base

Une entité exécutable est un programme qui s’exécute et qui respecte les quatre contraintes suivantes :

* Code exécutable chargé en mémoire
* Point d’entré : c’est l’adresse de l’instruction de démarrage (cette adresse doit être inscrire dans le descripteur de programme)
* Vecteur d’état : contient l’adresse de la prochaine instruction et le contexte du processeur
* Vecteur d’état doit être accessible à l’Unité Centrale.

Ils existent quatre types d’entités exécutables :

* Le processus : programme au sens courant du terme dont la structure d’affiliable est la file d’ordonnancement.
* Gestionnaire d’interruption : Surtout pour la gestion des entrée/sortie.
* Gestionnaire de déroutement : Pour gérer les anomalies (erreurs) du programme en cours d’exécution.
* Gestionnaire de communication : (semblable à la GI) est une communication qui se fait entre les processeurs

Le contexte de base : c’est l’ensemble des registres. Il en existe quatre familles regroupées en deux grandes familles : Registres généraux et registres spécialisés.

Les registres généraux (registre de données) utilisés comme des opérandes dans les instructions.

Les registres spécialisés (accessibles, cachés) :

* Registre accessible (Ex : compteur ordinal) : On peut lire et modifier son contenu.
* Registres cachés (Ex : Registre d’instruction) : On ne peut ni lire ni modifier son contenu.

Commutation de contexte : Le passage d’un programme à un programme suivant (Il s’agit de vider le registre et de permettre à l’entité élu d’occuper le processeur autrement dit Sauvegarder les registres puis les charger avec les données de l’entité élue. La commutation de contexte concerne les registres généraux et certains registres spécialisés).

Pour réaliser ce basculement, il y a deux scénarios possibles :

* La commutation de contexte déclenchée par le matériel :
  1. Sauvegarde du mot d’état (Compteur ordinal et le contexte du processus)[[1]](#footnote-1)
  2. Restauration du vecteur d’état de l’entité élue dans le mot d’état
  3. Sauvegarder les registres généraux[[2]](#footnote-2)
  4. Restaurer les registres généraux avec les données de l’entité élue.
* La commutation de contexte déclenchée par l’entité en cours :
  1. Sauvegarder son mot d’état
  2. Sauvegarder ses registres
  3. Restaurer les registres de l’entité élue
  4. Restaurés le mot d’état de l’entité élue avec son vecteur d’état.

## Ordonnancement des processus

La gestion de l’unité centrale entre les processus. Il se repose sur le descripteur de processus (est une structure de données créée par l’initialiseur et logé en mémoire).

Dans le descripteur on trouve :

* Vecteur d’état
* Zone de sauvegarde des registres
* Informations d’identifications : Numéro d’identification, Numéro du processus père, Numéro de groupe (Ex: pid et ppid en Lunix, Unix)
* Variable système : le temps d’exécution, les variables qui sont classés en deux grandes familles (les variables systèmes : l’adresse de l’instruction point d’entré, et les variables systèmes du processus)

Recherche : Comment définie une variable d’environnement.

* Les descripteurs mémoires : structure de données indiquant les espaces mémoires associés aux processus.
* Les descripteurs de ressources : les fichiers, un processus reçoit trois fichiers de la part de son père (fichier de sortir 2, fichier d’entré 0, fichier 1) et les fichiers de fenêtre (c’est une représentation en mémoire dont la copie est affichée à l’écran grâce à la carte graphique.
* Liens de chainage : Il en existe au moins un lien.

On a également les opérations d’ordonnancement

Ce sont des fonctions facilitant la commutation. On peut en citer :

* Arrêt par nécessité dans l’ordre : 1-3-4-2
* Préemptions, requisition (similaire à une interruption ou à un déroutement)
* Interruption / déroutement : 1-2-3-4.

# 04/11/2024

Mémoire : le lieu de transit des données.

Périphériques : c’est le matériel et l’électronique qui contrôle le périphérique (contrôleur de périphérique).

Pilote est soit un programme permettant de communiquer avec le contrôleur de périphérique ou encore le programme permettant de communiquer avec le matériel.

Bus sert de chemin de transit des données.

Le CP est vu comme de la mémoire par l’UC, offre une petite espace mémoire à l’UC qui ne constitue que des registres.

# Interaction entre le périphérique et UC

Il se réalise la plupart du temps par les interruptions.

Registre d’inhibition I -> registre R-> registre masque M -> Flag->registre de 8 et CC(génère un NI)

CPCI génère le NI à prendre en compte. En réalité on utilise les PIC pour gérer les Interruptions. Il est connecté à IM du processeur.

* Séquenceur : détecte l’interruption
* Séquenceur : récupère le mot d’état (Compteur ordinal et contexte de configuration du processeur), et l’empile dans la Pile d’exécution.
* Séquenceur : cherche le vecteur de l’interruption (calcule de l’adresse du vecteur de l’interruption = numéro du registre inscrits dans la table RTI + NI)
* Séquenceur charge du vecteur d’interruption dans le mot d’état.
* Séquenceur efface de l’interruption en mettant le flag a 0.

Pour exécuter le code du gestionnaire d’interruption, il faut :

* Masquer les interruptions
* Empiler les registres
* Transférer de l’unité de données suivantes, s’il en reste.
  + Sinon, réveillé le processus déclencheur du transfert en insérant son descripteur de processus dans la file d’ordonnancement
* Restaurer les registres
* Démasquer les interruptions
* IRET (Dépile le mot d’état qui a deux opérandes : la pile, le registre du mot d’état

# Exclusion Mutuelle 06/11/2024

## Problématique

Prenons une application P1 qui s’exécute en faisant une E/S en mode DMA En se moment si une autre application fait une demande d’E/S, elle devrait attendre la réveille de l’application P

## Resource

Tout ceux dont un programme à besoin pour être exécuter Elle est dite partager si elle peut être demander par deux programmes en même temps

**Exemple de ressources partagés** : UC, Mémoire, Bus, les périphériques, les variables et les structures de données situés à l’intérieure de la mémoire.

**Exemple TD** : partage d’une zone mémoire

## Code et séquence critiques

### Définition

Un code critique est un code qui utilise une ressource partagée

Une séquence critique est un code critique protégé

Si R est une ressource partagée critique alors une séquence critique sera :

Demander(R), Utiliser(R), Liberer(R) à l’intérieur Utiliser(R) est un code critique

Le but de l’exclusion mutuelle est protégé chaque code critique par un code d’exclusion mutuelle

Les techniques d’exclusion mutuelles :

* Un verrou : est une variable pour laquelle on a une instruction de lecture et écriture et L/E (Exemple : TAS/INTEL Test And Set) Lors de l’utilisation on suppose que R est libre si sa valeur est à 0 et peut être utiliser.

Ressource R , Verrou RV

Booléen b;

B=TAS(RV); :/\* Demande de la ressource) \*/

SI b == 0

Alors

Utiliser(R)

RV = 0 /\*Libérer R\*/

* Algorithme de Dekker
* Sémaphore : c’est la donnée d’un compteur C, d’une file d’attente de processus, et de deux fonctions P(S) demande et V(S) libération

## Fonctionnement de Sémaphore

Le compteur de C : Si C>=0 alors C == aux nombres d’exemplaires de la ressource disponibles.

Si c<0 => |c| == nombre de demandaires en attente = |FiIFO de s|

La séquence critique est P(RS), Utiliser(R), V(RS).

L’algo P d’un sémaphore

Algo de P

s.c = s.c – 1;

si(s.c < 0)

alors

Attendre(S.FiFO)

Algo de V

s.c = s.c+1

si(s.c <= 0)

alors

P=Retirer(S.FIFO)

insererFO(P)



P1:

P(RS)

E/S(R)

V(R.S)

P2()

P(R/S)

E/S(R)

V(RS)

# 08/11/2024 - Organisation et stockage des données

Les E/S sont représentées en 4 couches :

* **Niveau utilisateur** : ensemble des fonctions disponibles à être utiliser dans les programmes.
* **Niveau organisation** : pour la structuration des données sous formes de fichiers qui sont regroupés dans des répertoires. (copier, couper, déplacer, dupliquer, etc...)
* **Niveau stockage** : Les fonctions disponibles sont encapsulées dans les fonctions des deux niveaux précédents. Mais il y a une implémentation des fonctions permettant d’assurer l’implantations des fichiers dans le disque dur (Programmation des fonctions d’écriture) en gérant les adresses.
* **Niveau transfert** : gérer les échanges entre le périphérique et le processus par l’intermédiaire du contrôleur de périphérique.

Problèmes :

Un fichier est l’unité logique de stockage de l’information.

* Comment implémenter les fichiers (C’est quoi un fichier?)
* Comment banaliser l’accès aux fichiers.
* Gestion intuitive des fichiers 1,2
* Espace / Temps

Pour résoudre les problèmes, il faut mettre des fonctionnalités à disposition pour faciliter l’utilisation d’un fichier.

## Organisation de l’espace libre

Comment organiser l’espace pour implémenter les données ?

Pour le faire, on découpe le disque en blocs de 1Ko de façon logique. Cette taille est standard et non modifiable.

Comment distinguer un bloc libre et un bloc occupé ?

Un bloc est dit libre lorsque l’information contenue dans le bloc n’est pas significative.

* T1 Vecteurs de bits : On suppose que le premier bloc contient les informations sur les blocs de données.
* T2 listes chainées : Le premier bloc point sur le premier bloc libre et chaque bloc libre pointe sur le bloc libre suivant et l’insertion se fait en fin de chaîne. On peut aussi faire une liste chaînée des bloc occupés
* T3 Utilisation de table : Mais la table est utilisée pour gérer les blocs occupés

Gestion des espaces occupé par un fichier peut être géré selon trois techniques,

* Espace linéaire :
* Liste chaînée : il est difficile de modifier des éléments
* Tables des index : chaque fichier contient une table d’adresse (vecteur) dont le premier emplacement pointe sur le premier bloc de données

## Attribut et description d’un fichier

* Le nom du fichier
* Adresse
* Le numéro
* Le type
* La taille
* Les droits : lecture, écriture, exécution
* Les informations d’identification : les groupes
* Les dates : création, dernière modification.

## Système de fichiers

Les fonctionnalités d’E/S au niveau organisation. Dans windows, le système de gestion de fichiers est visualisé à travers l’explorateur. Dans Linux, on utilise des commandes.

De manière générale, un fichier contient trois types d’informations :

* Un descripteur de fichier : c’est une structure de données créer dans le premier bloc de données du fichier, on trouvera le type de fichiers, l’adresse du fichier (correspond à l’adresse de la Table des index), compteur d’utilisateur (Nombre d’application utilisant le fichier), Taille du fichier.

## Cause de la perte et de la fuite d’information

1. Effectué par le séquenceur [↑](#footnote-ref-1)
2. Effectué par l’entité élue [↑](#footnote-ref-2)