

Systèmes d'exploitation, 2ème année

Introduction générale

Yves STADLER

Université Paul Verlaine - Metz

13 septembre 2012

Contact



Yves Stadler

- A.T.E.R. IUT de Metz et doctorant au LITA ;
- Bureau B3.8 ;
- `yves.stadler@univ-lorraine.fr` ;
- `github.com/mvy/TC-INFO-ASR4-UPVM-YS`.

Agenda

Plan du module

- Gestion des processus
- Synchronisation et concurrence ;
- Communications ;
- Gestion de la mémoire ;
- Système de gestion des fichiers.

Présentation générale d'un système d'exploitation

Définition

Extension logicielle du matériel dans le but d'offrir un service suffisant aux utilisateurs.

Objectif

- Masquer la complexité à l'utilisateur ;
- Faciliter l'accès aux ressources.

Types de systèmes d'exploitations

- Mono-utilisateur, multi-utilisateurs.
- Mono-tâche, multi-tâches.
- Mode d'exploitation différé, interactif, temps partagé ;
- Mode temps réel, système embarqués ;

Présentation générale d'un système d'exploitation

Organisation d'un système d'exploitation

- Chaque système est basé sur un noyau (*kernel*) ;
- Le noyau comprend deux parties : indépendante du matériel, dépendante du matériel ;

Partie dépendante

- Gestion des interruptions ;
- Gestion mémoire ;
- Gestion des entrées sorties (E/S ; IO).

Présentation générale d'un système d'exploitation

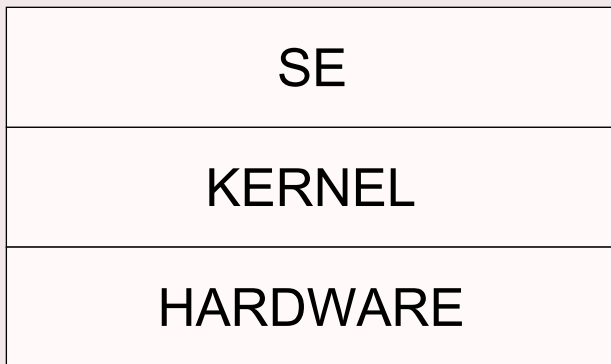
Partie indépendante

- Ordonnanceur-distributeur ;
- Gestion des processus ;
- Pagination, va-et-vient ;
- Sous-système de fichiers ;
- Gestion des entrées sorties (partie "haute").

Présentation générale d'un système d'exploitation

Modèle en couche

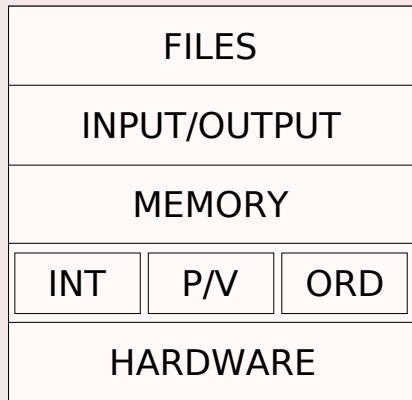
- Le système peut se présenter comme un ensemble de couches.



Présentation générale d'un système d'exploitation

Modèle en couche

- Le noyau ou *kernel* constitue une interface entre le matériel et les programmes
- Il est lui même divisé en couches
- Il est irremplaçable par l'utilisateur
- Monolithique, micro-noyaux ou hybrides



Présentation générale d'un système d'exploitation

Allocation de ressources

- Rôle central du *kernel* dans l'exécution des travaux :
 - représentation et gestion des processus ;
 - gestion des interruptions ;
 - gestion des entrées, sorties.

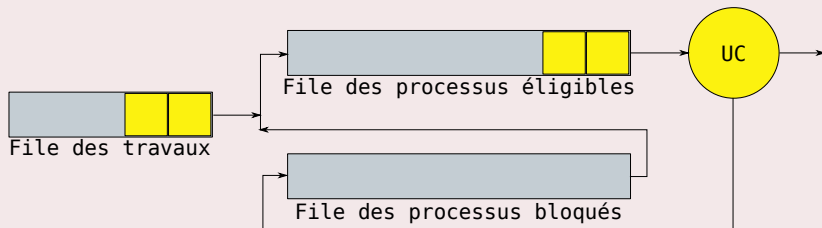


FIGURE : Exemple de modèle d'allocation de l'UC

Présentation générale d'un système d'exploitation



FIGURE : Allocation de l'UC pour les systèmes mono-tâche

Présentation générale d'un système d'exploitation

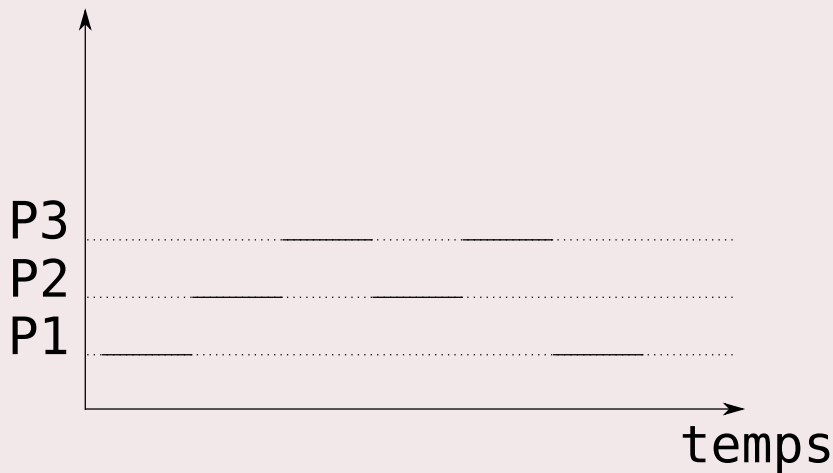
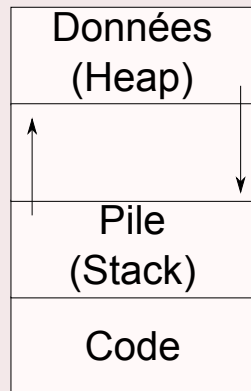


FIGURE : Allocation de l'UC pour les systèmes multi-tâches

Rappel sur d'architecture

L'espace de travail

- L'espace de travail est l'ensemble des données en mémoire nécessaires à l'exécution du processus.
- Le code : en langage ASM, liste des instruction pour le processeurs (Pointeur : Compteur Ordinal)
- La pile (stack) : mémoire prévue pour les appels de fonctions. Last In First Out.
- Les données statiques.
- Le tas (heap) : mémoire réservée pour l'allocation dynamique. Pas d'ordre particulier d'allocation. Données dynamiques.



Anticipation sur les processus

L'espace de travail

- Données privées du processus. Seule la zone `u` du processus en cours est manipulable. (`struct user <sys/user.h>`). Son adresse se trouve dans le mot état.
 - pointeur sur la structure de processus de la table des processus.
 - uid réel et effectif
 - compteurs des temps (users et system) consommés
 - masque de signaux
 - terminal de contrôle du processus si celui-ci existe.
 - dernière erreur rencontrée pendant un appel système.
 - valeur de retour du dernier appel système.
 - Entrées sorties (structures associées aux entrées-sorties)
 - "." et "/" (le répertoire courant et la racine courante (c.f. `chroot()`))
 - La table des descripteurs (fichiers ouverts)
 - Limites de la taille des fichiers de la mémoire utilisable
 - Umask (masque de création de fichiers)

Anticipation sur les processus

Contexte

- Son état
- Son mot d'état : en particulier (La valeur des registres actifs ; Le compteur ordinal)
- Les valeurs des variables globales statiques ou dynamiques
- Son entrée dans la table des processus
- Sa zone u
- Les piles user et system
- Les zones de code et de données.

Contexte

- Lorsqu'un nouveau processus va être exécuté, il y a commutation du mot d'état et changement de contexte. Ces changements sont dictés par l'ordonnanceur.

Mécanisme de contrôle

Mécanismes

- Les interruptions ;
- Les appels systèmes ;
- Les signaux horloge ;
- Les primitives de synchronisation.

Les interruptions

- Événement suffisamment important pour nécessiter une interruption du système.
- Rapide, immédiat.
- Générée aléatoirement par un périphérique ou par l'UC ;
- Peut être générée par le système en interne, on parle alors de détournement.

Mécanisme de contrôle

Rôle des interruptions

- Imposer les changements d'état de l'UC ;
- Commutation de contexte, générée par une cause extérieur à l'instruction en cours.

Conditions d'interruption

- L'UC doit être en mode interruptible ;
- L'interruption doit être prioritaire aux autres interruptions ;

Mécanisme de contrôle

Appel au superviseur - Définition

- Instruction qui a pour effet de provoquer une commutation de contexte du processeur.

Appel au superviseur - Rôle

- Permettre l'appel depuis un programme d'une procédure du système nécessitant des droits étendus.
- Masquage interruption, allocation de mémoire ...

Mécanisme de contrôle

Signaux d'horloge

- Composant physique du système ;
- Essentiel car il rythme le système ;
- Génère des interruptions horloges.

Signaux d'horloge

- Affectation du temps dans le mot horloge
- Pendant l'activation du processus, le mot horloge est décrémenté de 1 à chaque signal ;
- Quand le mot vaut 0, un signal d'interruption est généré et le superviseur prend une décision.

Mécanisme de contrôle

Primitives de synchronisation

- Nécessaires à l'asynchronisme ;
- Nécessaire à la protection mutuelle ;
- Plus que de simples appels de procédures.

Exemple de l'exclusion mutuelle

- Deux processus A et B veulent mettre à jour le compte en banque d'Alice
- $C_{Alice} = C_{Alice} + Montant_a$ et $C_{Alice} = C_{Alice} + Montant_b$
- Quel est le résultat final ?