Systèmes d'exploitation (OS) Cheat Sheet

by solsolsol via cheatography.com/32070/cs/9820/

Process

Un process est un programme en cours d'exécution.

Ressources nécessaires:
- ⟶ Temps cpu
- ⟶ Memoire
- ⟶ Fichiers et peripheriques I/O

Responsabilités du système:
- ⟶ Créer et effacer les processus
- ⟶ Mise en veille/re-activer process
- ⟶ Fournir des mécanismes pour:
- ▯ > Synchroniser process
- ▯ > Faire communiquer les process

Mémoire principale

Liste de cases mémoire sous forme de bytes ou de words (2bytes) référencés par une adresse.

Data pool à accès rapide partagé entre le CPU et périphériques I/O.

Péripherique volatile: Data perdu en cas de panne du systeme.

Responsabilités du système:
- ⟶ Grader une trace de:
- ▯ > Occupation mémoire
- ▯ > Qui l'utilise
- ⟶ Décide:
- ▯ > Next process load in mem
- ▯ > When free mem. space available
- ⟶ Memory allocation on demand

Mémoire secondaire (morte) (DD)

Stockage programmes et données.

Accès **lent** mais grande capacité.

Responsabilités du système:
- ⟶ Gestion de l'espace libre
- ⟶ Allocation de l'espace
- ⟶ Gestion accès disques

Fichiers (files)

Groupe d'informations du même genre.
Contient:
- ⟶ Programme (code source/compilé)
- ⟶ Données

Fichiers (files) (cont)

Responsabilités du système:
- ⟶ Make/delete files
- ⟶ Make/delete folders
- ⟶ Fournir des primitives
- ▯ (fonction bas niveau) pour la manipulation
- ▯ de fichiers et de repertoires
- ⟶ Make link with secondary memory
- ⟶ Sauvegarde les fichiers dans une mémoire
- ▯ non-volatile (Secondary memory: DD...)

Système distribué

Collection de PC autonomes connectés via protocole de communication. Chaque poste exécute des composantes coordonnées avec celle des autres.

L'utilisateur perçoit le système comme un unique système intégré.

Exemple: Programme SETI (Search for Extra-Terrestrial Intelligence). Séquences de calculs, issues du découpage d'un projet de calcul global.

Accès à des ressources partagées:
- ⟶ Augmentation vitesse calcule
- ⟶ Augmentation disponibilité des données
- ⟶ Augmentation de la fiabilité

I/O (Entrées/Sorites)

- ⟶ Système de cache avec mémoire tampon
- ⟶ Interface de gestion générique
- ⟶ Logiciel pour accès au matériel spécifique

Système de protection

Mécanismes pour contrôler l'accès aux ressources du système par des processus systèmes ou des programmes utilisateurs.

- ⟶ Détermine si l'accès est autorisé ou non
- ⟶ Spécifie les contrôles à imposer
- ⟶ Fournit des moyens de les faire respecter

Interprétation de commandes

Les commandes données à l'os doivent gérer:
- ⟶ La création/destruction de process
- ⟶ Les I/O
- ⟶ La mémoire secondaire (morte, DD)
- ⟶ La mémoire principale (vive, RAM)
- ⟶ L'accès au système de fichiers
- ⟶ La protection
- ⟶ Le réseau

Commande Language Interpreter (CLI)(W)

Shell (UNIX)

Sa fonction est d'obtenir prochaine commande et l'exécuter.

Services des OS

Exécution de programmes: Charger un programme, en mémoire et l'exécuter)

Operations I/O: Pour des raisons d'efficacité et de sécurité, user ne les contrôle pas directement. L'OS fournit services pour accès I/O.

Manipulation des systèmes de fichiers: Lire, écrire, créer, éffacer et modifier des fichiers.

Communication: Echange d'informations entre plusieurs processus dans le même système ou dans des systèmes séparés reliés par un réseau.

Détection d'erreurs: Assure un fonctionnement en détectant les erreurs dans CPU, mémoire, I/O ou dans les programmes user.

Appels Système

Fournissent l'interface entre process et OS:
- ⟶ Généralement en language Assembleur
- ⟶ Language de remplacement pour accès OS

Trois méthodes sont utilisées pour transmettre des paramètres à l'OS:
- ⟶ Passage par registres
- ⟶ Memorisation des paramètres dans un
- ▯ tableau et transmission de l'adresse du
- ▯ début du tableau par un registre
- ⟶ Empiler (push) des paramètres sur la pile
- ▯ qui seront ensuite "dépiles"(pop) par l'OS

Types d'appels systèmes:

Cheatography

Systèmes d'exploitation (OS) Cheat Sheet

by solsolsol via cheatography.com/32070/cs/9820/

Appels Système (cont)

Contrôle de processus:
→ Load,exec,stop,make,wait,warn,allocate
Gestion de fichiers:
→ Make,delete,open,close,read,write,move
Gestion des périphériques:
→ Request,free,read,write,move
Maintenance de l'information:
→ Récupérer/changer date et heure
→ Gérer les données système pour les process, les fichiers ou les périphériques
Communications
→ Créer ou détruire les connexions
→ Gérer les messages

Fonctions supplémentaires

Ensemble de fonctions pour assurer un comportement correct de l'OS.

Allocation de ressources entre plusieurs utilisateurs et processus fonctionnant simultanément sur le même système.

Comptabilisation des ressources utilisées par chaque utilisateur de processus (utilisé pour des statistiques ou de la facturation).

Protection accès aux ressources système.

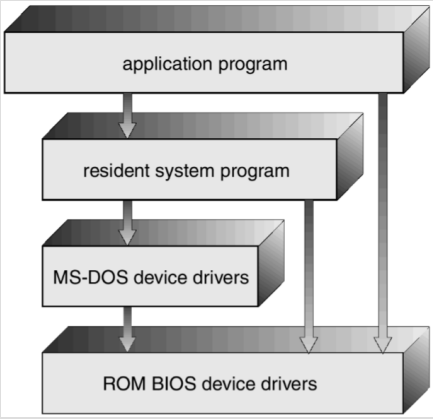
Structure systèmes MS-DOS

Ecrit pour donner max fonctionnalités dans un minimum d'espace. Ecrit "rapidement".

Decoupage des modules bâclé.

Interfaces et niveaux mal séparés.

Structure des couches MS-DOS



Structure système UNIX

Système original: Structure simple car limité par les fonctionnalités matériel.

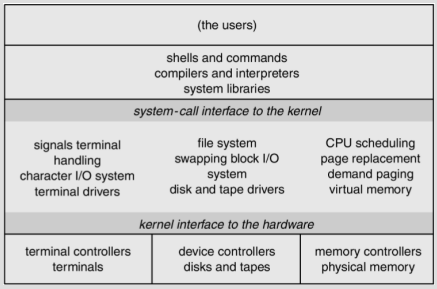
Décomposition en deux parties distinctes:
→ Programmes système
→ Noyau (Kernel)

Kernel:

Contient toutes les fonctions de gestion depuis les appels système jusqu'au matériel.

Fournit le système de fichiers, l'ordonnancement des process, gestion mem + autres fonctions bas niveau.

Structure systémes UNIX



OS divisé en couches

Chaque niveau s'appuie sur le niveau inf. Le niveau le plus bas est constitué par le matériel, le plus haut est l'interface user.

L'organisation des couches doit être approprié. Chaque couche ne doit utiliser que des fonctions de la couche inférieure.

Machines Virtuelles

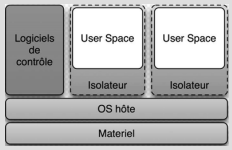
Le matériel et le noyau sont traités comme une seule couche matérielle.

Une MV simule un matériel en fournissant une interface identique.

L'OS crée l'illusion d'avoir plusieurs process fonctionnant en même temps chacun sur son processeur avec sa propre mémoire virtuelle.

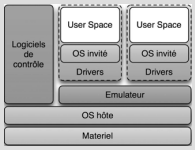
Ressources physiques du PC sont partagées pour créer des MV:
→Ordonnancement CPU crée l'illusion que chaque user a son propre CPU
→Simulation possible de périphériques

Isolateur (Software)



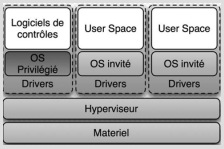
Isole l'exec des apps dans Zone d'exécution.
Fait tourner plusieurs fois la même app même si elle n'était pas conçue pour ça.
Très performant mais +- != Virtualisation car le virtualenv n'est pas complètement isolé.
cLinux-Vserver,BSD Jail, OpenVZ

Hyperviseur Type 2



Permet de lancer un ou plusieurs OS guest.
Le pc maître émule le matériel pour les OS guest, qui croient dialoguer directement avec ledit matériel.
Couche physique émulée, Isolation complète mais perfs réduites /rapport à la virtualisation.
Echanges entre OS (Guests, Master) vi canaux standards de la com entre OS (TCP/IP...). Un tampon d'échange permet d'émuler des cartes réseaux sur une seul CR réelle. **ex: VirtualPC, VirtualBox...**

Hyperviseur Type 1



Guest accède direct à la couche physique
Para-virtualisation: Guest sait qu'il est virtualisé et donc + opti, + de perfs.
ex:VMware vSphere, MS Hyper-V Server,...