**Modules :** Décomposer un système informatique en un ensemble de modules (ou « services ») 将计算机系统分成一系列模块的集合

分布式操作系统 SX

Chaque module joue un rôle, réalise un service précis (bien identifié) et spécifique (pas de recoupement/doublon entre modules).

Chaque module est caractérisé par une interface, qui propose une abstraction du service réalisé. 接口是模块实现功能的抽象

Conception descendante( 自上而下的设计概念） : Le système informatique est conçu comme une hiérarchie (une succession) de services

（将模块组合起来） Simplifie (abstrait) la machine précédente et/ou ajoute une fonction à la machine précédente

（通过逻辑总线） reliés par un bus logiciel, qui gère les interactions entre services.

**Interface** = ensemble des fonctions fournies aux «clients» du service. Chaque fonction est définie par **son format (sa syntaxe)** et **sa spécification (sa sémantique)**

\*L'interface est **séparée**, distincte de la réalisation (implémentation) du service 其特点是

- portabilité ; - maintenance ; - standardisation ; (l'interface reste stable, les réalisations peuvent changer selon le contexte)

- protection : l'interface est un passage obligé pour l'accès au service (encapsulation)

**Fonctions d’un SX : \*** Comment simplifier l’accès aux ressources matérielles ? 如何简化对物质资源的获取 ? 虚拟化

Proposer une interface simplifiée d’accès aux ressources

masquant les détails de mise en œuvre

éliminant les contraintes physiques des ressources réelles

offrant des opérations plus évoluées, abstraites

Implanter cette interface.

\* Un SX présente en général **deux (types d’) interfaces :操作系统通常有两种接口**

Interface utilisateur(用户接口), ou interface de commande(命令接口)

\* **Interface programmatique** (API) : (编程接口)

Fournit un environnement d’exécution abstrait (simplifié) aux programmes dont l’exécution est gérée par le système

\* **E/S synchrones** : 为了减少CPU与IO的时间réduire le temps consacré par l’UC à la gestion des échanges avec les périphériques (E/S)

Utiliser des périphériques rapides ! temps d’E/S réduit

Préparer les E/S : transférer les données des périphériques lents vers les périphériques rapides de manière indépendante de l’UC

(Hiérarchie de mémoires et,E/S « virtuelles») 将IO的数据准备到读取速度较快的设备里

\* **E/S asynchrones(异步IO)** : CPU ne gère pas les E/S ;(cpu不管理io设备)

CPU et processeurs périphériques partagent un tampon mémoire(cpu和外设共享内存缓冲区)

Communication entre CPU et périphériques indépendants(cpu和外设的通信): 轮询**scrutation** & 中断**interruptions**

虚拟机的概念 \* notion de **machine virtuelle** :

Chaque traitement en cours (processus) dispose de son environnement d’exécution : ensemble de ressources nécessaires

le SX gère l’état d’allocation des ressources aux processus :

par processus : ressources attendues et ressources obtenues | par ressource : processus élus et processus en attente

encapsulation des ressources par un arbitre (noyau/superviseur) : ayant seul directement accès aux ressources ;

disposant d’une vue globale de l’état du système ; accès aux ressources ! appels au noyau

分布式系统 \* Partage de ressources (serveurs) : impression, stockage -> architectures client/serveur (C/S架构)

communication entre utilisateurs : outils de travail coopératif

Intégration du parallélisme dans le modèle utilisateur用户的并行 : parallélisme «utilitaire» (pour le SX) -> parallélisme comme service

SX实现机制

进程的概念 **processus** (dynamique) : activité exécution d’un programme (statique) par un processeur.

进程 & 程序 Un programme peut être exécuté par plusieurs processus en même temps, chaque processus travaillant sur ses propres donnée

Répartir à tout instant les ressources entre les différents processus, de sorte à : permettre l’exécution simultanée de plusieurs processus ; fournir à chaque processus les ressources nécessaires à son exécution始终在不同进程之间分配资源, 以便允许同时执行多个进程, 为每个流程提供执行所需的资源

Mémoire centrale : Une partie de la mémoire espace système est attribuée au SX. Cet espace contient notamment les données utiles à la gestion

des processus (descripteurs de processus), des ressources (requêtes en attente), des échanges avec les périphériques (tampons d’E/S. . . )

**Partage temporel 中断机制** : (1) 中断interrompre le processus en cours et sauvegarder son état ;

(2) 恢复restaurer l’état du processus suivant, puis reprendre son exécution

**Contexte** 概念 : Contexte est un ensemble des informations à sauvegarder pour pouvoir poursuivre un traitement interrompu entre 2 instructions.

une instruction indivisible 不可分割的指令 : **Sauvegarde du contexte courant** 保存当前; **Restauration du prochain contexte**恢复下一个

**进程的状态**Etat d’un processus : valeur des données utilisées ; code exécuté ; état (contexte) du processeur.

切换进程的方法/ OS与IO的交互Interaction entre le SX et les périphériques : Attente active (**scrutation**)轮询 ; **Interruptions** 中断(mécanisme matériel).

中断**Interruptions**的流程 : 系统无需等待le SX n’attend pas, mais poursuit son activité ;

事件发生时外设通知系统 un périphérique signale au SX le moment où l’événement survient ;

系统中断当前活动,处理中断 à la réception du signal, le SX interrompt le traitement en cours pour traiter l’événement signalé ; 处理后恢复源环境 l’événement traité, le SX reprend le traitement interrompu.

**Scrutation & Interruption :** 轮询 simple et fiable ; coûteux et inadapté aux applications temps-réel

中断 réponse « rapide » au « signal » de Pi ; plus complexe ; requiert un support matériel (ITs + commutation)

中断等级Niveaux d’interruption : causes d’interruption possibles, identifiées 已确定的可能中断原因

Chaque niveau peut être traité séparément, par une routine (procédure) de traitement (traitant, handler)

Les adresses de ces routines sont souvent conservées dans une zone fixée de l’espace système : vecteur d’interruption 中断向量

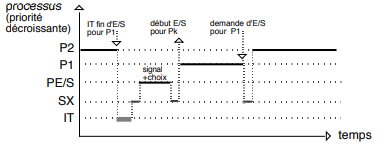
Priorités possibles entre niveaux

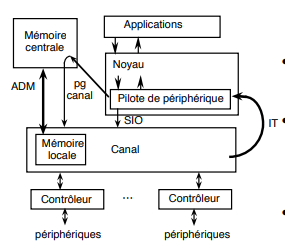
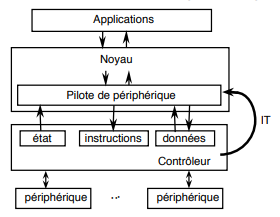
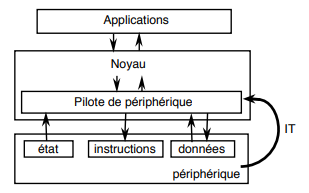
**Masque d’interruption** : ensemble des niveaux pouvant provoquer une IT.

**Niveau désarmé** : l’IT est ignorée **Niveau masqué** : le traitement de l’IT est retardé

当一个IT已经在处理中, 而收到另一个IT时会发生什么？Que se passe t-il lorsqu’une IT est reçue alors qu’une autre IT est déjà en cours de traitement ?

Selon le matériel, ou le niveau, la nouvelle IT peut être ignorée (IT désarmée) ; provoquer l’abandon du traitement d’IT en cours (IT perdue)

 être traitée selon une politique de niveaux de priorité.



程序员视角 : Activité d’exécution d’un programme par un processeur. Processus = suite d’actions = suite d’états obtenus = trace.

进程 Processus

机器状态Etat de la machine = état du processeur (registres) + état de la mémoire (données)

执行指令Exécution d’une instruction = changement d’état de la machine 改变机器状态

执行程序Exécution d’un programme = exécution d’une suite d’instructions 一系列指令的执行

操作系统视角 : Le SX doit gérer le partage des ressources, afin que chaque processus finisse par disposer des ressources nécessaires à son exécution.

Processus = utilisateur de ressources -> état d’un processus = état d’allocation des ressources

L’activité d’un processus est abstraite (réduite) aux opérations d’allocation/restitution de ressources 进程=资源分配or归还的操作

état d’un processus = état par rapport à l’allocation des différentes ressources 与各种资源分配相关的状况

Environnement d’exécution : processeur et mémoire vive sont gérés entièrement par le SX -> transparent pour l’utilisateur/le programmeur cpu和ram由os管理

fork() : Création d’un processus fils : fork() Hérite de l’environnement construit par le processus père. Exécute **le même programme** que le père

exec() : Commutation de programme : exec() Le fils charge un nouveau programme à exécuter该子加载另一个新的程序(与父进程不同)运行

La table des processus -(regroupe)-> Les descripteurs de processus(PCB) <-(conserver)- Les informations d’allocation des ressources

**Descripteur de processus :** identifiant du processus ; copie du contexte processeur (à la dernière commutation) ;

informations de gestion des ressources ; liens vers les processus créateur (père)/créés(fils)

Pour **Unix** : Descripteur de processus = **Structure utilisateur** + **Structure de processus**

**Le contexte du processus :** (进程描述符 + 数据结构) comprend le descripteur de processus et les structures qu’il référence (tables, piles. . . )

进程操作的流程 : 1.Création(创建) : (1)Création du descripteur de processus initialisé (2)Le processus est prêt ou suspendu

2.Destruction/Terminaison (释放): (1)Libération des ressources utilisées par le processus (2)Libération du descripteur de processus

**Ordonnancement des processus** : (目的) Gérer l’allocation des ressources aux processus

进程调度 (1) une file est associée à chaque ressource, qui contient les (descripteurs des) processus en attente de la ressource

(2) lorsque la ressource est disponible,le choix du prochain processus auquel allouer la ressource dépend de la nature de la

ressourceet de critères définissant la politique d’allocation appliquée

**Allocation du processeur** : Court terme : **priorité(优先权)** associée aux processus, ou non ; **réquisition(抢占)** (préemption), ou non

进程分配调度策略 FIFO : Principe原理 : servir les requêtes par ordre chronologique d’arrivée. 简单, 无优先权, 无抢占

不适合时分 inadapté au temps partagé

Tourniquet (Round-Robin) : Principe原理 : durée maxi. d’activité continue pour le processus actif (élu)

les processus prêts sont rangés dans une file >> le processeur est alloué au processus en tête de file

>> le processeur est réquisitionné (préempté) au profit du processus suivant dans la file

与FIFO相似 ; 适合时分adapté au temps partagé

Allocation à deux niveaux : Principe : Les processus prêts sont divisés en catégories (processus interactifs, tâches de fond. . . )

Chaque catégorie a sa politique d’allocation (FIFO, tourniquet...)

Une politique d’allocation est définie entre catégories(FIFO, tourniquet, priorités. . . )

Ordonnancement temps-réel : Garantir aux traitements le respect decontraintes de dates « précises » de terminaison (échéances)

ordonnancement à taux monotone (RMS : Rate Monotonic Scheduling)

échéance la plus proche (EDF : Earliest Deadline First)

**Périphériques blocs :** Données structurées (regroupées) dans des blocs (souvent de taille fixe)

(**块设备**,如磁盘) Chaque bloc est identifié par une adresse ; L’accès aux données est déterminé par le numéro du bloc

**périphériques caractères (字符设备**,如键盘): Données non structurées, vues comme des suites (flots) d’octets ; L’accès aux données doit suivre l’ordre du flot

**Caches** : 目标optimiser les échanges en conservant dans un tampon (cache) en mémoire rapide une copie des données échangées

**Fichier**文件概念 : Fichier = une suite **d’octets** et une suite **d’enregistrements**

Abstraction pour un ensemble de données数据集的抽象

Conservées indépendamment de l’exécution des applications (rémanence) 独立保存于应用程序的执行

Ayant un lien logique (pour l’utilisateur) 具有逻辑链接（针对用户

Différenciées par un identifiant (clé) ou un indice (position) 通过标识符（键）或索引（位置）来区分

**后缀名.xx** Convention : il est pratique d’ajouter au nom un suffixe **indiquant la nature du contenu (指示内容的性质).**

**Protection et partage des fichiers** : confidentialité : informations accessibles aux seuls usagers autorisés机密性：仅授权用户可以访问的信息

intégrité : pas de modifications non désirées完整性：无意外修改

contrôle d’accès : seuls certains usagers sont autorisés à faire certaines opérations访问控制：仅某些用户有权执行某些操作

authentification : garantir qu’un usager est bien celui qu’il prétend être身份验证：确保用户就是他声称的那个人

打开文件的步骤 : (1) The inode number of the file found by the system. (2) Find the inode information of the file according to the inode number

(3) Find the file and open it according to the inode information

**Structure type d’un disque logique**: 1.文件名 2.inode 3.block

| Bootblock(磁盘结构,文件系统等) | Superblock(存放inode号码) | Inode Table(存放文件的inode信息) | File(真正存放文件的地方) |

**Partitions** (disques virtuels) : L’interface avec l’organisation physique.

分区 Unix propose 2 types de disques logiques : swap : la régulation et la gestion mémoire systèmes de fichiers : stockage des fichiers

**Implantation des partitions** : bloc d’amorce : code d’amorce + table des partitions

分区布局 table des partitions : taille et adresse des différentes partitions réserve de blocs

**Représentation de l’espace libre** :Vecteur de bits : pour chaque bloc du disque, un bit indique si le bloc est libre 对于每个磁盘块,用1bit来表示这个块是否空闲

可用空间的表示形式 优点Avantages : simple et efficace. 缺点Inconvénient : à condition que le vecteur tienne en mémoire centrale

Liste chaînée : L’espace libre est vu comme une suite de blocs. Chaque bloc libre contient l’adresse du suivant.

缺点Problème : Inefficace (allouer n blocs -> n E/S)

**Implantation des fichiers sur le disque** : Allocation contiguë : Chaque fichier est rangé dans des blocs d’adresses successives 存储在连续的地址块中

磁盘上文件的位置 L’entrée du catalogue pour le fichier contient l’adresse du premier bloc et la taille du fichier

**Avantages**优点 : permet l’accès direct et l’accès séquentiel ; temps d’accès minimal, pour un fichier

**Inconvénients**缺点 :**Fragmentation externe**外部碎片

Allocation fragmentée : Pas de relation entre les adresses (la localisation) des blocs du fichier.

Table d’allocation **(FAT)** est indexée par les numéros de blocs disque. 分配表由磁盘块号索引

le numéro du bloc suivant ; 0 = le bloc est libre ; EOF = le bloc est le dernier du fichier.

Allocation chaînée : Les blocs alloués à un fichier sont chaînés. Chaque bloc contient l’adresse de son successeur.

**Avantages** : pas de fragmentation externe ; pas de problème lié à l’évolution de la taille d’un fichier.

**Inconvénients** : inefficace 低效; perdre un bloc 数据丢失

Allocation indexée : Le premier bloc de chaque fichier est la table des adresses des blocs alloués au fichier

**Avantages** : 同Allocation chaînée ; Accès direct possible

**Inconvénient**: perte de place pour les petits fichiers 解决办法如下

**Méthode hybride** (Unix) : l’entrée du catalogue pour un fichier(i-nœud) contient 15 adresses de blocs : 12 adresses directes (48K adressables) 3 adresses indirectes

**I-nœud UNIX：**| mode (bloc/caractère) | propriétaires (user, groupe) | date (modification, etc.) | taille | nb de blocs | adresses directes (12) | adresses indirection(3) |

**Interface utilisateur** pour la manipulation des **fichiers** : 提供的服务permettre noms externes ; structurer l’espace de noms/d’objets défini par l’utilisateur

目录的定义Notion de **répertoire** : Ensemble de fichiers nommés et regroupés au choix de l’utilisateur

**Arborescence树形结构** : inclure des répertoires dans les répertoires.

Intérêt : organisation définie par l’utilisateur ; contrôle des droit d’accès

Règle les conflits de noms (un fichier est désignable par le chemin (unique) le reliant à la racine)

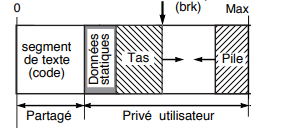
**chemin absolu** 绝对路径 : Chemin reliant la racine au fichier. **chemins relatifs** 相对路径: utilisation d’un préfixe implicite (répertoire de travail/courant

**Liens physiques** (nom interne) (commande *ln f ./f\_bis*) : ajout d’une entrée *f\_bis* au répertoire courant, associée au même i-nœud que *f*

硬链接 : 以文件副本存在, 但不占用实际空间, 目录répertoire不能硬链接

Liens symboliques (chemin d’accès) (*ln -s f ./f\_bis*) : le chemin du fichier *f* est stocké dans le fichier *f\_bis* du répertoire courant.

软链接 : 以路径形式存在, 类似于快捷方式, 可对目录répertoire链接.

Pour le programmeur : image mémoire = zone mémoire (RAM)

Privée私人的

Contiguë连续的

Commençant à l’adresse 0从地址0开始

Bornée par la taille des adresses受地址大小限制

**Allocation de blocs de mémoire contiguë 连续分区:**

**Partitions fixes** : (1) La mémoire est divisée en zones de taille fixe (partitions). (2) Une file d’attente est associée à chaque partition

**(固定分区)** (3) Quand un processus est lancé, le SX lui alloue la plus petite partition pouvant contenir son image mémoire (best fit)

Avantage : efficace Problème : **fragmentation interne内部碎片**

**Partitions variables** : Allouer à chaque processus une partition de la taille exacte de son image mémoire. En fin d’exécution, l’espace occupé est libéré

**(可变分区)** Problème : **fragmentation externe 外部碎片** Method : **(Re)compactage :** regrouper les zones libres en un seul bloc

**Fragmentation externe 外部碎片 :** L’espace libre est suffisant, mais divisé en nombreux fragments de petite taille, inutilisables séparément (miettes (garbage))

**Fragmentation interne内部碎片 :** Internal fragmentation means that the memory space that has been allocated is larger than the memory space required by the

request, and the memory space that cannot be used is internal fragmentation.

**Allocation fragmentée**零散分区Pagination (Allocation fragmentée de zones fixes)

Idée : fractionner l’image mémoire d’un processus >> gérer une carte d’implantation de l’image mémoire (table des pages)

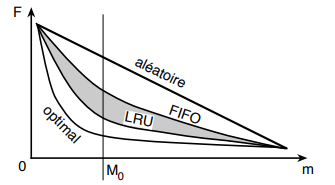
L’image mémoire (virtuelle) est découpée en pages. La mémoire physique est découpée en cases. Cases et pages ont une taille P fixée et identique

Av= (n° de page,déplacement dans la page) = (p,d). p = Av对p取整, d = Av对d取余

**Avantages** : pas de fragmentation externe ; fragmentation interne : 1/2 page par processus, en moyenne

**Segmentation**段 : Mémoire vue comme un ensemble de zones (segments) de longueurs variables ; sans ordre relatif

Pour chaque segment : contrôle des accès ; contrôle de l’adressage

**Algorithmes de remplacement** : Algorithme aléatoire : Victime = page choisie aléatoirement

FIFO : Victime = première page chargée (parmi les pages présentes)

Algorithme optimal : Victime = page dont la prochaine utilisation est la plus tardive

LRU(Least Recently Used): Victime = page inutilisée depuis le plus longtemps

Point saillant : existence d’un seuil M0 à partir duquel F croît très vite

avant ce seuil, les performances s’améliorent nettement avec m

après ce seuil, augmenter m est peu rentable