

Chapitre 01:

Généralité sur les systèmes d'exploitation

(RTOS: Real Time Operating System)

1. Définitions:

Un **système d'exploitation (OS)**: est un ensemble de programmes spécialisés qui permet l'utilisation des ressources matérielles d'un ou plusieurs ordinateurs de façon optimale et efficace. Un OS fournit une interface homme-machine (IHM) permettant la communication entre l'utilisateur et les machines par les différents logiciels d'application.

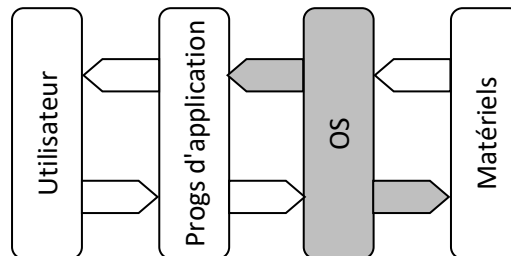


Figure 1.1. L'emplacement d'un OS.

2. Rôles d'un OS:

- **Gestion du processeur** : il gère l'allocation du processeur entre les différents programmes grâce à un algorithme d'ordonnancement.
- **Gestion de la mémoire vive** : il gère l'espace mémoire alloué à chaque application. En cas d'insuffisance de mémoire physique, le système d'exploitation peut créer une zone mémoire sur le disque dur, appelée «mémoire virtuelle».
- **Gestion des entrées/sorties** : il permet d'unifier et de contrôler l'accès des programmes aux ressources matérielles par l'intermédiaire des pilotes (gestionnaires de périphériques).
- **Gestion de l'exécution des applications.**
- **Gestion des fichiers** : il gère la lecture et l'écriture dans le système de fichiers et les droits d'accès aux fichiers par les utilisateurs et les applications.
- **Gestion des informations.**

3. Composantes des systèmes d'exploitation:

3.1. Le noyau (kernel) toujours en RAM d'OS, il contient les fonctionnalités critiques du SE: elles doivent:

- toujours être prêtes à l'utilisation
- traitement d'interruptions
- gestion d'UCT
- gestion de mémoire, fichiers, processus, des entrées sorties principales
- communication entre processus ... etc.

3.2. Interpréteur de commande/coquille (shell) permettant la communication avec le système d'exploitation par l'intermédiaire d'un langage de commandes, afin de permettre à l'utilisateur de piloter les périphériques.

3.3. Système de fichiers (file system), permettant d'enregistrer les fichiers dans une arborescence.

4. Notions importantes:

Processus: Suite continue d'opérations, d'actions constituant la manière de faire, de traiter quelque chose.

Programme: Ensemble d'instructions et de données représentant un algorithme qui peut être exécuté par un ordinateur.

Instruction: une opération élémentaire d'un processeur dans une architecture d'ordinateur.

Macro-instruction: instruction complexe, définissant des opérations composées à partir des instructions du répertoire de base d'un ordinateur.

Un système d'exploitation multitâche (*multithreaded*): lorsque plusieurs tâches (processus) peuvent être exécutées simultanément.

Un système préemptif: lorsqu'il possède un **ordonnanceur (planificateur)**, qui répartit, selon des critères de priorité, le temps entre les différents processus qui en font la demande (les interruptions).

Le multiprocessing: une technique qui consiste à faire fonctionner plusieurs processeurs en parallèle afin d'obtenir une puissance de calcul plus importante ou bien afin d'augmenter la disponibilité du système (en cas de panne d'un processeur). Un

système multiprocesseur gère le partage de la mémoire entre plusieurs processeurs, également il distribue la charge de travail.

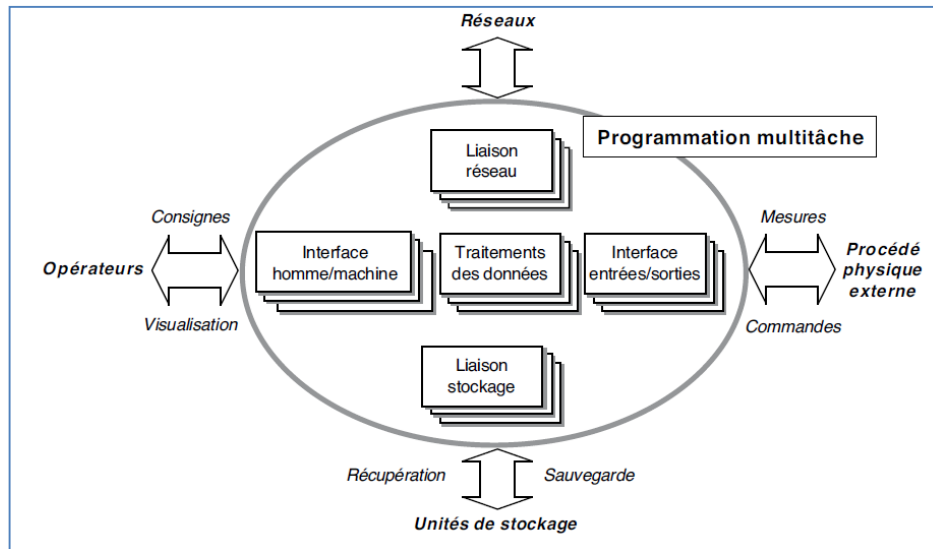


Figure 1.2. Architecture logicielle d'une application de contrôle-commande multitâche.

5. Types des OS:

5.1. OS généralistes:

- Windows
- Linux (Fedora, Debian, Ubuntu, etc...)

5.2. OS spécialisés:

- Android
- BlackberryOS
- iOS
- OSEK/VDX (Automobile)

5.3. OS temps réel: fonctionne de manière fiable selon des contraintes temporelles spécifiques, c'est-à-dire qu'il doit être capable de délivrer un traitement correct des informations reçues à des intervalles de temps bien définis. Voici quelques exemples de systèmes d'exploitation temps réel :

- RTLinux (hard Real Time Linux, USA);
- QNX (la société BlackBerry, USA/Canada);
- Xenomai (2001, USA);
- OpenVMS (la société hp, USA, utilisé dans le projet "la ligne 14 métro Paris");

- Micro-Itron (libre, Japan).
- RTEMS, FreeRTOS (libres et légers).

Remarque : Linux n'est pas TR, il peut être modifié pour être Temps Réel (PREEMPT-RT, Xenomai).

6. Différences entre OS et RTOS :

6.1. OS (Système d'exploitation ou système opératoire) :

1. Au démarrage, l'OS prend le contrôle,
2. Compilation + Edition des liens + Exécution du programme,
3. Multiprogrammation, multi-utilisateurs.

6.2. RTOS (Système d'exploitation temps réel) :

1. Edition des liens de l'application et du RTOS,
2. Au démarrage, l'application prend le contrôle et démarre le RTOS,
3. RTOS et l'application sont fortement couplés,
4. Aucune protection vis-à-vis de l'application → meilleure performance,
5. Services limités aux Systèmes embarqués → réduction de taille mémoire,
6. Configuration du RTOS : gestion des fichiers, pilotes des E/S, gestion de la mémoire, Outils,...

7. Fonction d'un RTOS :

Les principales fonctions d'un système d'exploitation pour la commande de processus en temps réel sont les suivantes:

- 1-** l'action sur les dispositifs externes (convertisseurs, lecteurs de capteurs, contrôleurs de vannes, etc.);
 - 2-** la prise en compte du temps réel, en fournissant toute réponse dans un temps minimum;
 - 3-** la réaction aux événements extérieurs, avec le minimum d'interventions humaines;
 - 4-** la gestion fiable des informations permettant un fonctionnement, même en cas de défaillances matérielles.
-

8. Exemples de quelques RTOS dans l'industrie :

8.1. RTEMS (Real-Time Executive for Multiprocessor Systems) est un RTOS libre pour les systèmes embarqués (<http://www.rtems.com>). Il permet de développer des programmes dans lesquels le temps de réponse et la réactivité sont des contraintes fortes (temps réel dur). Il a été porté sur de nombreux processeurs (ARM, i386, m68k, MIPS, PowerPC) et est très complet.

8.1.1. Ses Caractéristiques :

1. Conforme à la norme POSIX1003.1b (API).
2. Système multitâches incluant les multi-activités (threads).
3. Supporte les systèmes multiprocesseurs homogènes ou non.
4. Supporte le pilotage par événement.
5. Ordonnancement préemptif basé priorités.
6. Supporte l'ordonnancement RM (rate monotonic scheduling).
7. Communication et synchronisation entre tâches.
8. Héritage de priorités.
9. Gestion des réponses aux interruptions.
10. Supporte l'allocation dynamique de la mémoire.
11. Pile TCP/IP.

8.2. VxWorks : est aujourd'hui l'exécutif temps réel le plus utilisé dans l'industrie. VxWorks est fiable, généralement utilisé dans les systèmes embarqués et porté sur un nombre important de processeurs (PowerPC, 68K, ColdFire, MCore, 80x86, Pentium, i960, ARM, StrongARM, MIPS, SPARC, NECV8xx,...).

Un point fort de VxWorks a été le support réseau (sockets, NFS,...) dès son apparition. VxWorks est également conforme à POSIX1003.1b.

8.2.1. Ses Caractéristiques :

1. Noyau TR multitâches préemptif avec ordonnancement round robin.
 2. Faible temps de latence.
 3. Mémoire protégée permet d'isoler les applications utilisateurs du noyau.
-

4. Support des multiprocesseurs.
5. Communication inter-tâches.
6. Implémente l'exclusion mutuelle par sémaphores.
7. Héritage de priorités.
8. Gestion de file de messages locaux et distribués.
9. Système de fichiers.
10. Pile TCP/IP IPV.
11. Latence moyenne de 1.7 ms, latence max de 6.8 ms sur Pentium 200 MHz (<http://www.windriver.com>).

8.3. QNX : est un RTOS adapté aux applications critiques. Développé par QNX Software Systems, sa structure est de type Unix et il est compatible POSIX.

On le retrouve entre autres dans le développement de logiciels, le contrôle de robots industriels et les ordinateurs embarqués. Son micro-noyau Neutrino peu gourmand en ressource confère à QNX des capacités temps-réel très performantes avec un temps de latence très faible (0.55 μ sec sur un Pentium III). Il dispose d'une interface graphique nommée Photon (version 4.x). Le temps de commutation de tâches est très faible (<1 μ sec).

8.3.1. Ses Caractéristiques :

1. Supporte la norme POSIX 1003.1b.
2. Micronoyau Neutrino.
3. Zone mémoire protégée (Run time Memory protection).
4. Multitâches préemptif.
5. Micro-GUI Photon.
6. Supporte les processeurs x86, SMP jusqu'à 8 processeurs.
7. Système de fichiers.
8. version actuelle : 6.4 (<http://www.qnx.com/>).

8.4. RTLinux : se présente sous la forme d'un micronoyau prenant place entre la machine réelle (matériel) et le noyau Linux. Il intègre un gestionnaire de tâches (qui dans la pratique est un module). Le noyau de Linux partage le temps processeur avec les autres tâches temps réel et devient une tâche du RTOS.

Il est utilisé pour des produits sensibles dans le domaine des télécommunications ou des applications militaires. RTLinux propose des systèmes de communication de type FIFO ou mémoire partagée entre les tâches temps réel (dans l'espace noyau) et l'espace utilisateur dans lequel s'exécutent habituellement les applications. Initialement produit libre, il est maintenant devenu un produit commercialisé par la société FSMLabs (<http://www.fsmlabs.com>).

8.4.1. Ses Caractéristiques :

1. Système préemptif, ordonnanceur à priorités fixes.
2. Supporte plusieurs processeurs (SMP).
3. Supporte les architectures x86, PPC Alpha (DEC), MIPS, StrongARM, ...
4. Autorise la communication et synchronisation IPC (Inter Process Communication : tubes, files de messages, sémaphores, mémoire partagée,...).
5. Communication entre processus utilisateurs et noyau (tâches temps réel) par FIFO, mémoire partagée ou signaux.
6. Noyau Linux est alors une tâche de faible priorité (idle task).