

Introduction aux systèmes d'exploitation, les processus, l'ordonnancement

Luigi Santocanale

Laboratoire d'Informatique Fondamentale,
Centre de Mathématiques et Informatique,
39, rue Joliot-Curie - F-13453 Marseille

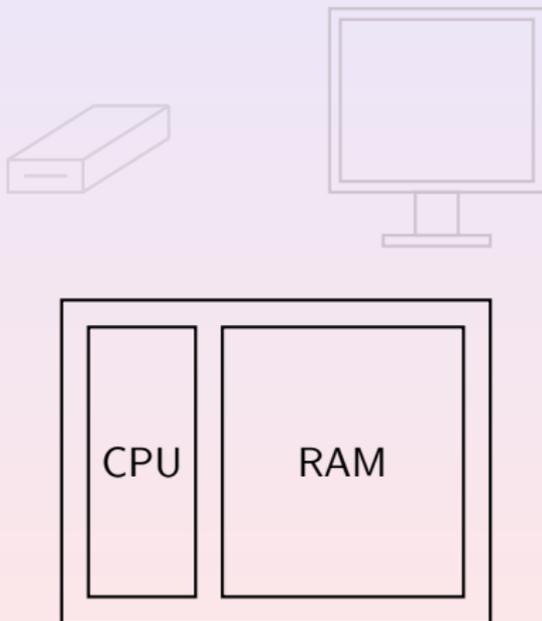
25 septembre 2005

- 1 Introduction
 - Que est-ce qu'un système d'exploitation
 - Un peu d'histoire
 - Introduction à Unix
- 2 Introduction aux processus
 - Généralités
 - États d'un processus
- 3 Éléments d'ordonnancement
 - Généralités
 - Ordonnancement avec réquisition (systèmes temps partagée)

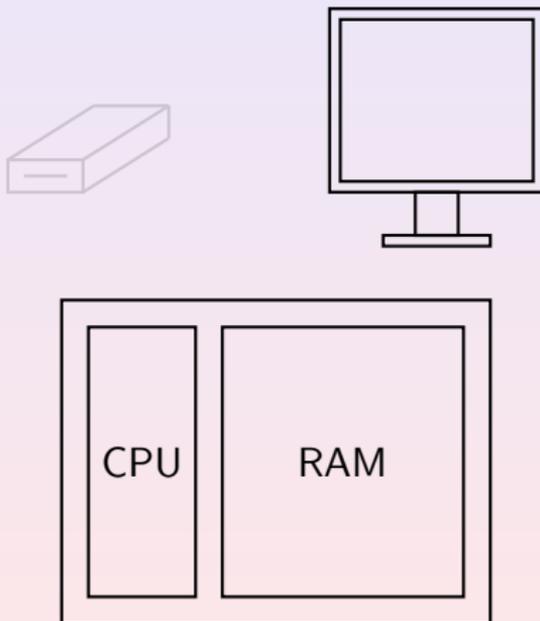
Plan

- 1 Introduction
 - Que est-ce qu'un système d'exploitation
 - Un peu d'histoire
 - Introduction à Unix
- 2 Introduction aux processus
 - Généralités
 - États d'un processus
- 3 Éléments d'ordonnancement
 - Généralités
 - Ordonnancement avec réquisition (systèmes temps partagée)

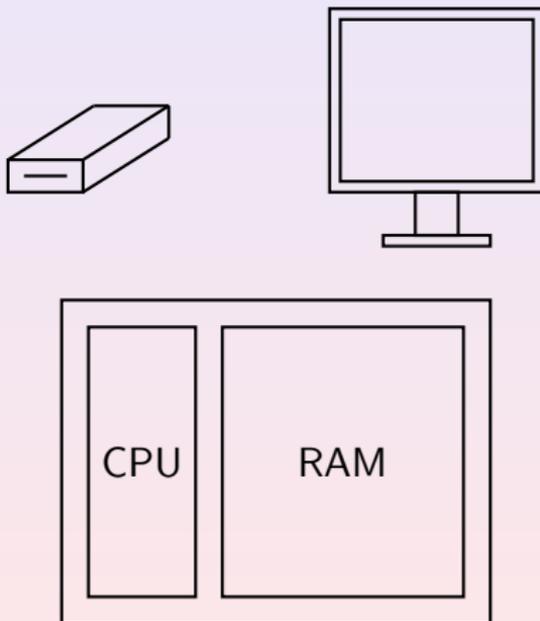
OS : machine virtuelle



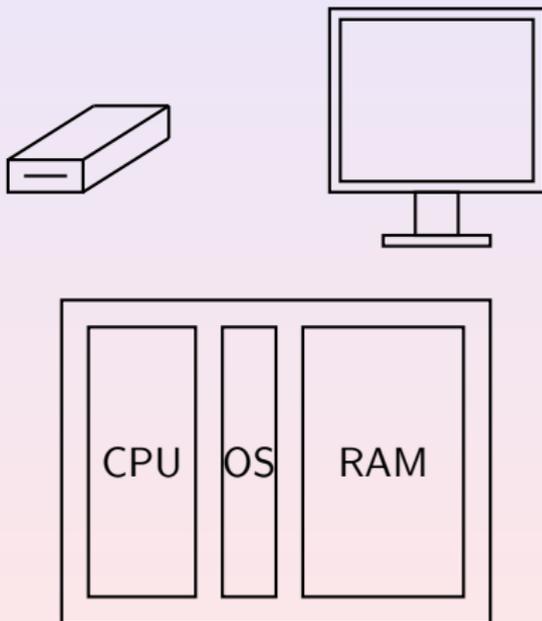
OS : machine virtuelle



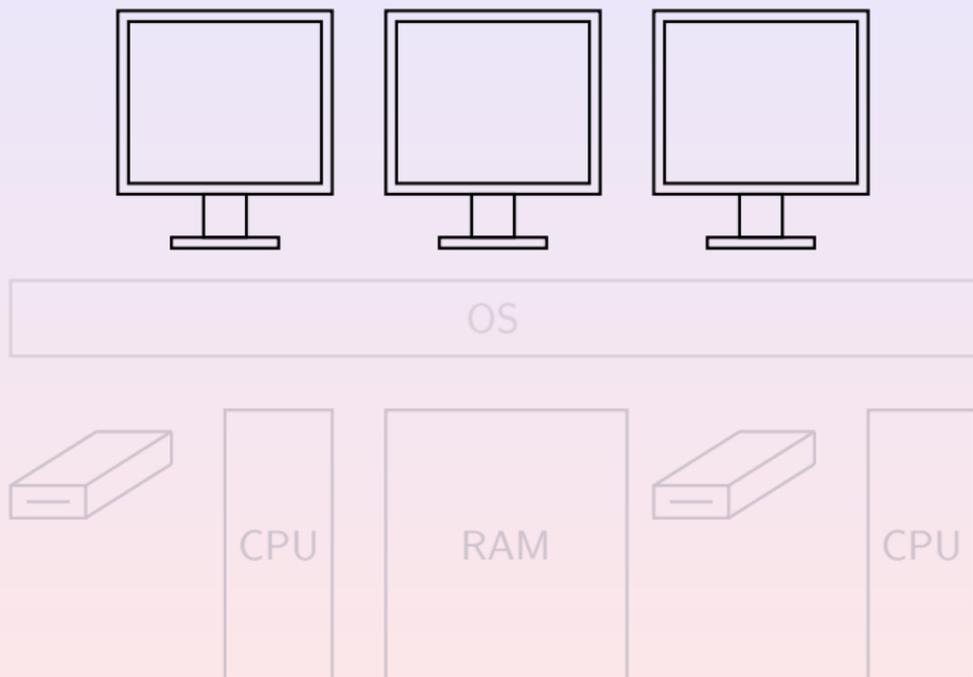
OS : machine virtuelle



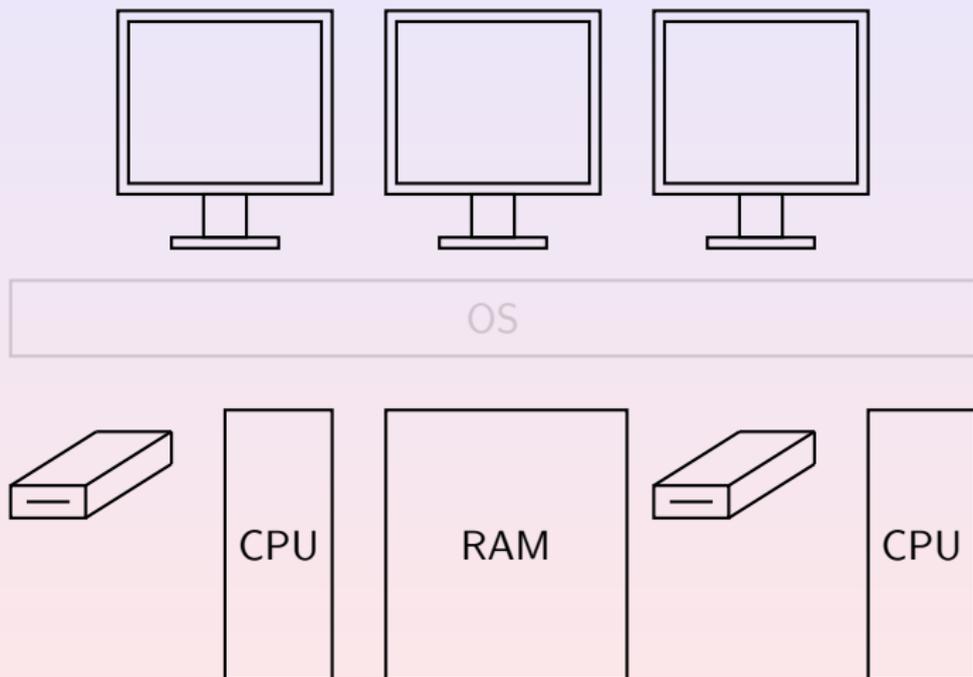
OS : machine virtuelle



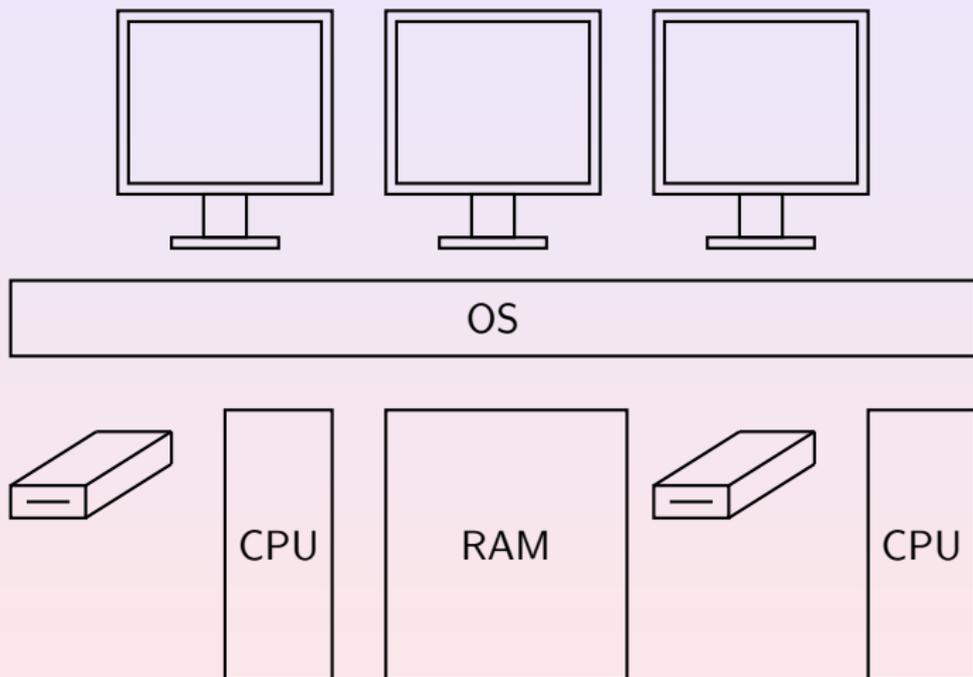
OS : allocateur de ressources



OS : allocateur de ressources



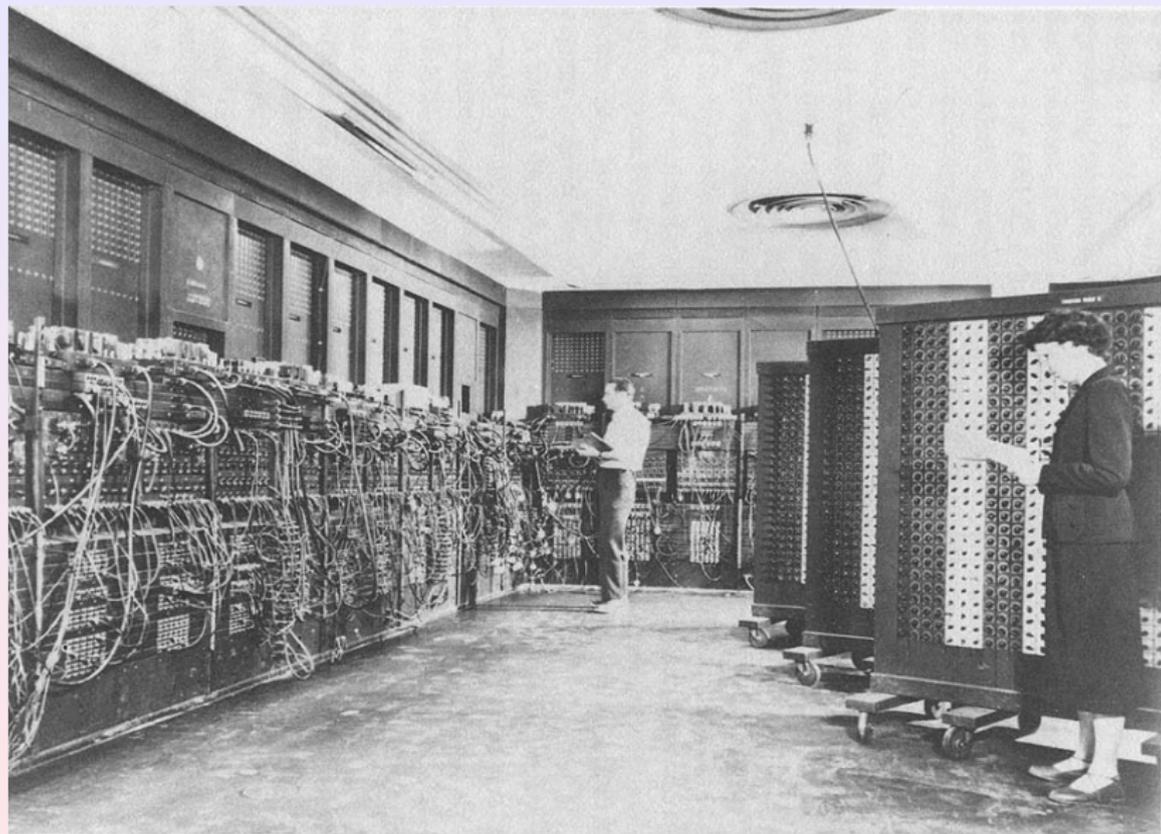
OS : allocateur de ressources



Plan

- 1 Introduction
 - Que est-ce qu'un système d'exploitation
 - **Un peu d'histoire**
 - Introduction à Unix
- 2 Introduction aux processus
 - Généralités
 - États d'un processus
- 3 Éléments d'ordonnancement
 - Généralités
 - Ordonnancement avec réquisition (systèmes temps partagée)

1945 – 1955 : les tubes à vide



The ENIAC.

Smithsonian Institution Photo No. 53192.

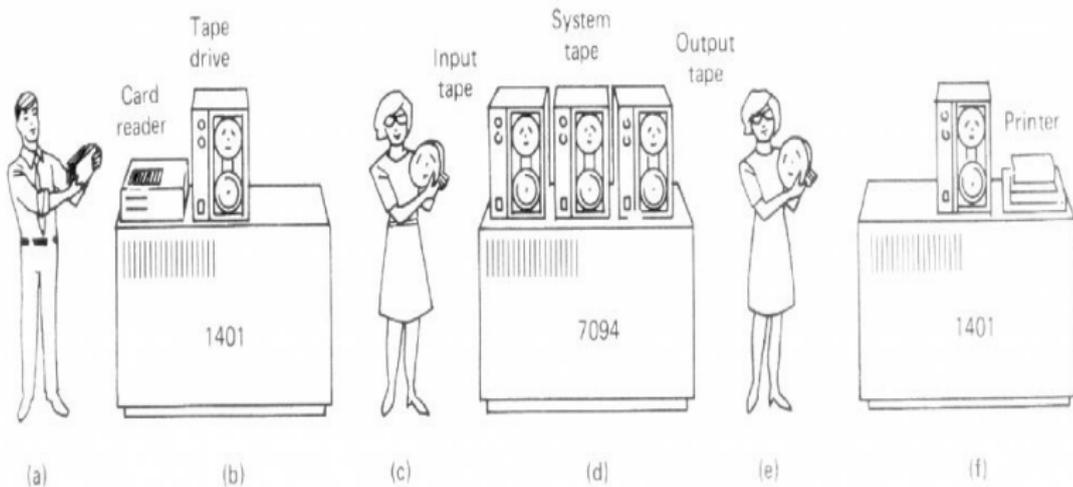
1945 – 1955 : les tubes à vide

ENIAC : 20000 tubes à vide, 160m²

- constructeur = programmeur = utilisateur
- lots d'heures allouées au programmeur
- programmation par câblage,
pas de langages de programmation
- pas de mémoire
- calculs : tables des sinus et cosinus

1955 – 1965 : les transistors

- mainframes IBM 7094
- écriture du source sur carte perforé
- appui de « petit » ordinateurs IBM 1401 pour collectionner les taches sur bande magnétique
- opérateur humain transfère les bandes magnétiques vers/de l'ordinateur



- traitement d'un ensemble de travaux
- transfert sur bande magnétique
- opérateur insère et récupère les bandes
- l'ordinateur compile et charge automatiquement
- impression off-line

1965 – 1980 : Circuits intégrés

- IBM system 360 : famille d'ordinateurs homogènes
- calcul scientifique vs. calcul commerciale :
le pb des E/S
- la multiprogrammation :
 - plusieurs taches en mémoire
 - mise en attente des jobs bloqués en E/S
- spooling (Simultaneous Peripheral Operation On Line) :
chargement automatique des nouvelles taches (jobs)

Limites : manque d'interactivité

Le partage du temps

- On découpe le temps en tranche
- Les utilisateurs interagissent sur des terminaux
- MIT, Bell Labs, General Electric :
développement de MULTICS.
Modèle : le système de distribution de l'électricité

1980 – ... : les ordinateurs personnels

- Circuits LSI (Large Scale Factors) :
baisse de prix.
- Mini-ordinateur → Micro-ordinateur : chacun a son ordinateur.
- Développement d'applications conviviaux, « User-friendly » :
destinés à l'utilisateur sans diplôme en informatique.
- MS-DOS pour IBM 8088, UNIX pour Motorola 68000.
- Versions ultérieures de MS-DOS intègrent des éléments de UNIX.

Plan

- 1 Introduction
 - Que est-ce qu'un système d'exploitation
 - Un peu d'histoire
 - Introduction à Unix
- 2 Introduction aux processus
 - Généralités
 - États d'un processus
- 3 Éléments d'ordonnancement
 - Généralités
 - Ordonnancement avec réquisition (systèmes temps partagée)

Unix : historique

Unics

1969 : portage de MULTICS sur un PDP-7
par Ken Thompson chez Bell Labs

1970 : portage de UNICS sur un PDP-11
arrivée de D. Ritchie, réécriture en C,
portage de UNIX sur autres machines

1977 : développement à l'université de Berkeley

1983 : faillite de AT&T,
droits commerciaux sur System V

1984 : norme IEEE 1003

1985 : didactique : UNIX minimal

1991 : implantation de UNIX sur les PCs

Unix : historique

Unics
|
Unix

1969 : portage de MULTICS sur un PDP-7
par Ken Thompson chez Bell Labs

1970 : portage de UNICS sur un PDP-11

arrivée de D. Ritchie, réécriture en C,
portage de UNIX sur autres machines

1977 : développement à l'université de Berkeley

1983 : faillite de AT&T,
droits commerciaux sur System V

1984 : norme IEEE 1003

1985 : didactique : UNIX minimal

1991 : implantation de UNIX sur les PCs

Unix : historique

Unics
|
Unix

1969 : portage de MULTICS sur un PDP-7
par Ken Thompson chez Bell Labs

1970 : portage de UNICS sur un PDP-11
arrivée de D. Ritchie, réécriture en C,
portage de UNIX sur autres machines

1977 : développement à l'université de Berkeley

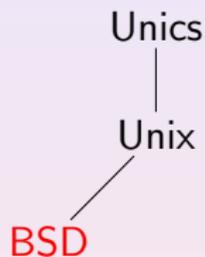
1983 : faillite de AT&T,
droits commerciales sur System V

1984 : norme IEEE 1003

1985 : didactique : UNIX minimal

1991 : implantation de UNIX sur les PCs

Unix : historique



1969 : portage de MULTICS sur un PDP-7
par Ken Thompson chez Bell Labs

1970 : portage de UNICS sur un PDP-11
arrivée de D. Ritchie, réécriture en C,
portage de UNIX sur autres machines

1977 : développement à l'université de Berkeley

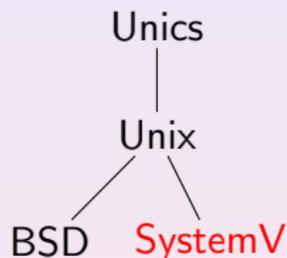
1983 : faillite de AT&T,
droits commerciales sur System V

1984 : norme IEEE 1003

1985 : didactique : UNIX minimal

1991 : implantation de UNIX sur les PCs

Unix : historique



1969 : portage de MULTICS sur un PDP-7
par Ken Thompson chez Bell Labs

1970 : portage de UNICS sur un PDP-11
arrivée de D. Ritchie, réécriture en C,
portage de UNIX sur autres machines

1977 : développement à l'université de Berkeley

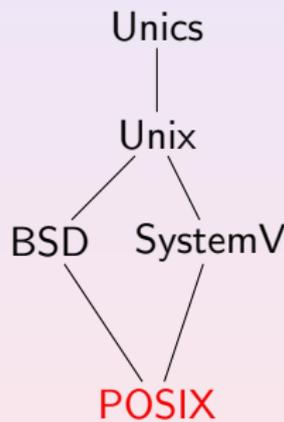
1983 : faillite de AT&T,
droits commerciales sur System V

1984 : norme IEEE 1003

1985 : didactique : UNIX minimal

1991 : implantation de UNIX sur les PCs

Unix : historique



1969 : portage de MULTICS sur un PDP-7 par Ken Thompson chez Bell Labs

1970 : portage de UNICS sur un PDP-11
arrivée de D. Ritchie, réécriture en C, portage de UNIX sur autres machines

1977 : développement à l'université de Berkeley

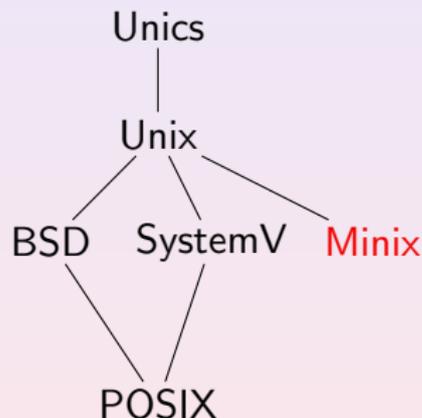
1983 : faillite de AT&T, droits commerciaux sur System V

1984 : norme IEEE 1003

1985 : didactique : UNIX minimal

1991 : implantation de UNIX sur les PCs

Unix : historique



1969 : portage de MULTICS sur un PDP-7 par Ken Thompson chez Bell Labs

1970 : portage de UNICS sur un PDP-11
arrivée de D. Ritchie, réécriture en C, portage de UNIX sur autres machines

1977 : développement à l'université de Berkeley

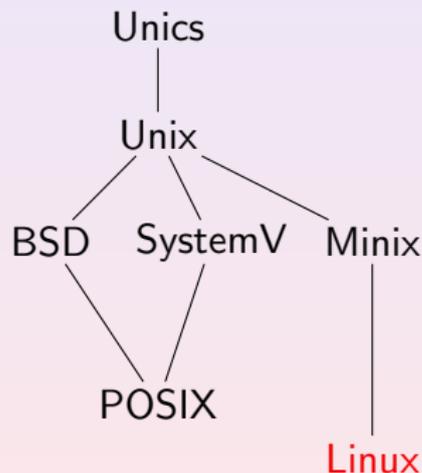
1983 : faillite de AT&T, droits commerciaux sur System V

1984 : norme IEEE 1003

1985 : didactique : UNIX minimal

1991 : implantation de UNIX sur les PCs

Unix : historique



1969 : portage de MULTICS sur un PDP-7 par Ken Thompson chez Bell Labs

1970 : portage de UNICS sur un PDP-11
arrivée de D. Ritchie, réécriture en C, portage de UNIX sur autres machines

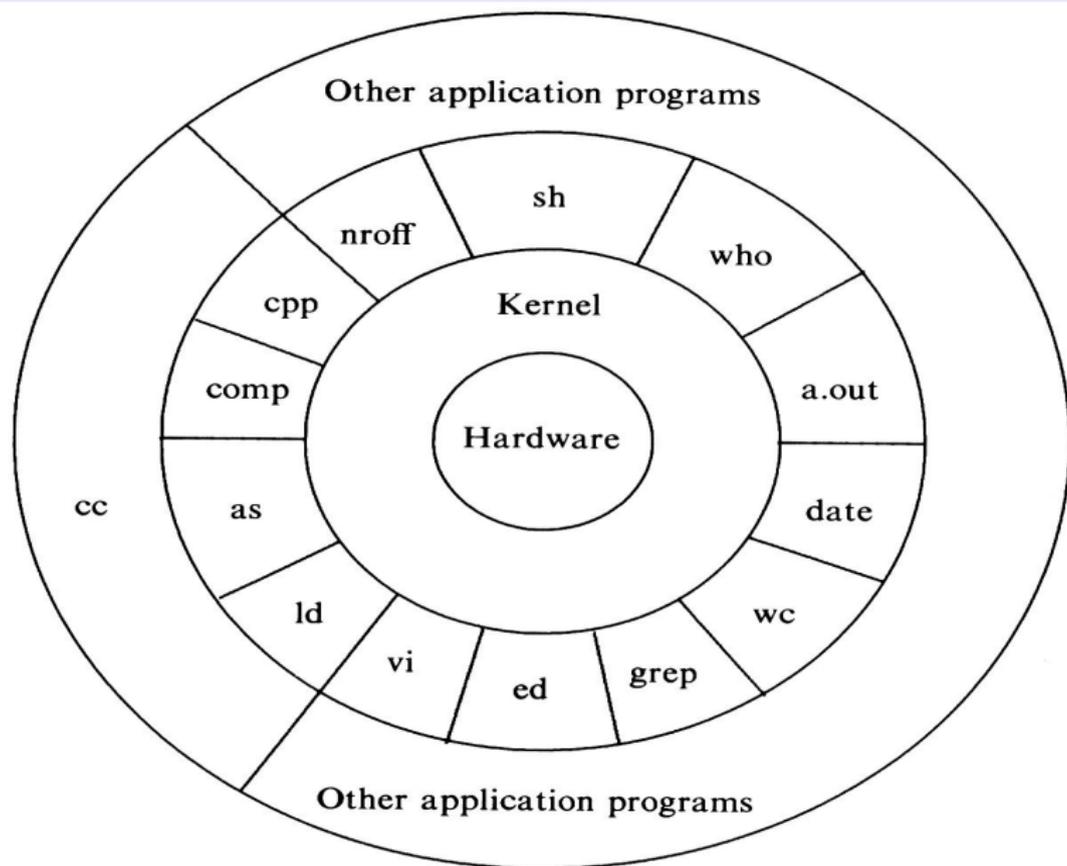
1977 : développement à l'université de Berkeley

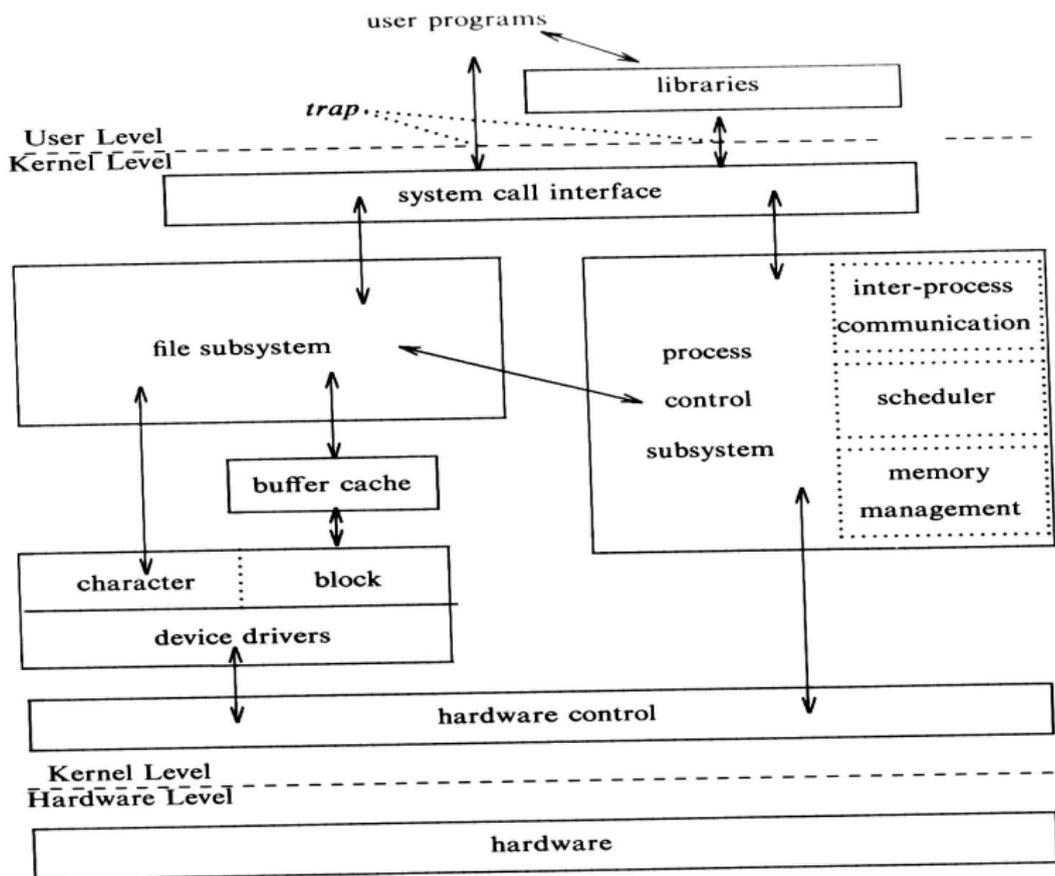
1983 : faillite de AT&T, droits commerciaux sur System V

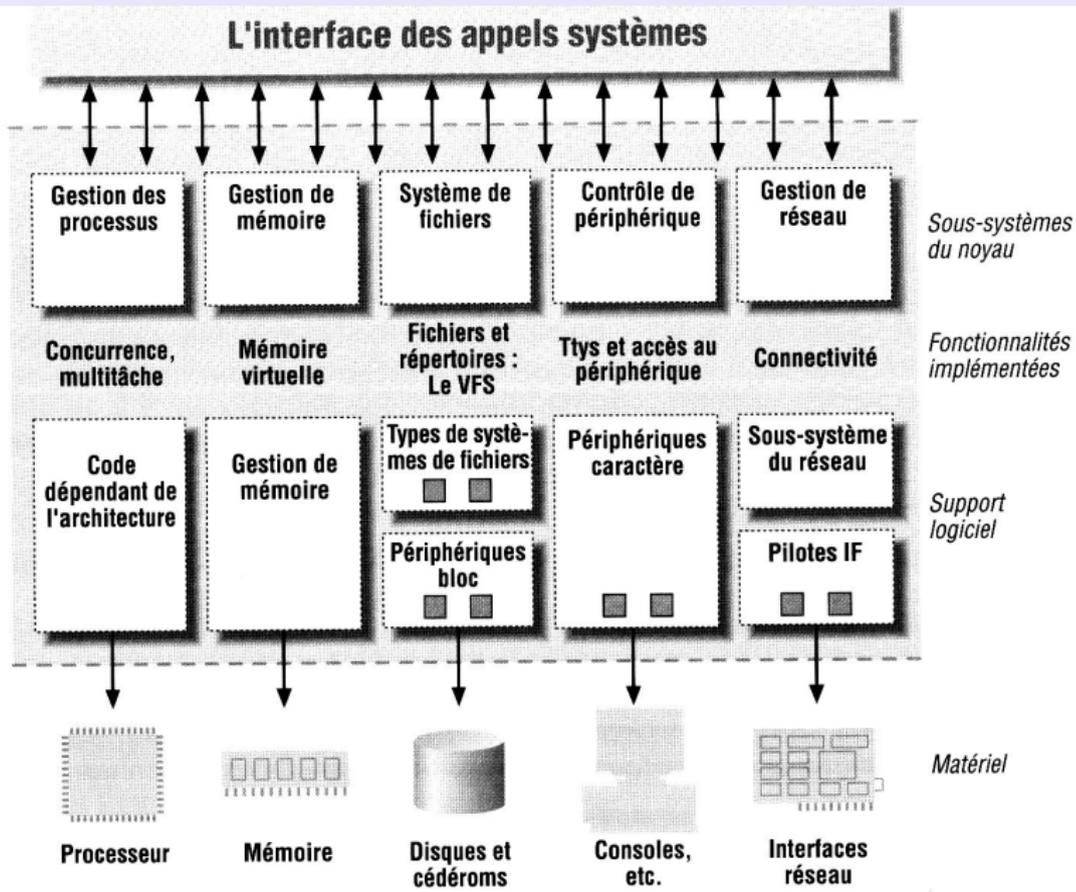
1984 : norme IEEE 1003

1985 : didactique : UNIX minimal

1991 : implantation de UNIX sur les PCs







Étudier le systèmes d'exploitation UNIX ?

- 1 Utilisateur :
savoir utiliser le système UNIX.
- 2 Administrateur-programmeur :
savoir gérer un système UNIX : programmation shell-script.
- 3 Programmeur :
savoir bien utiliser les appels système (interface POSIX) dans un programme.
- 4 Théorie :
les problèmes posés par la conception d'un système d'exploitation.
- 5 Développeur de systèmes:
problèmes posés par l'implantation d'un système d'exploitation.

Ici: on s'occupera de 3,4 et 5.

Étudier le systèmes d'exploitation UNIX ?

- 1 Utilisateur :
savoir utiliser le système UNIX.
- 2 Administrateur-programmeur :
savoir gérer un système UNIX : programmation shell-script.
- 3 Programmeur :
savoir bien utiliser les appels système (interface POSIX) dans un programme.
- 4 Théorie :
les problèmes posés par la conception d'un système d'exploitation.
- 5 Développeur de systèmes:
problèmes posés par l'implantation d'un système d'exploitation.

Ici: on s'occupera de 3,4 et 5.

Étudier le systèmes d'exploitation UNIX ?

- 1 Utilisateur :
savoir utiliser le système UNIX.
- 2 Administrateur-programmeur :
savoir gérer un système UNIX : programmation shell-script.
- 3 Programmeur :
savoir bien utiliser les appels système (interface POSIX) dans un programme.
- 4 Théorie :
les problèmes posés par la conception d'un système d'exploitation.
- 5 Développeur de systèmes:
problèmes posés par l'implantation d'un système d'exploitation.

Ici: on s'occupera de 3,4 et 5.

Étudier le systèmes d'exploitation UNIX ?

- 1 Utilisateur :
savoir utiliser le système UNIX.
- 2 Administrateur-programmeur :
savoir gérer un système UNIX : programmation shell-script.
- 3 Programmeur :
savoir bien utiliser les appels système (interface POSIX) dans un programme.
- 4 Théorie :
les problèmes posés par la conception d'un système d'exploitation.
- 5 Développeur de systèmes:
problèmes posés par l'implantation d'un système d'exploitation.

Ici: on s'occupera de 3,4 et 5.

Étudier le systèmes d'exploitation UNIX ?

- 1 Utilisateur :
savoir utiliser le système UNIX.
- 2 Administrateur-programmeur :
savoir gérer un système UNIX : programmation shell-script.
- 3 Programmeur :
savoir bien utiliser les appels système (interface POSIX) dans un programme.
- 4 Théorie :
les problèmes posés par la conception d'un système d'exploitation.
- 5 Développeur de systèmes:
problèmes posé par l'implantation d'un système d'exploitation.

Ici: on s'occupera de 3,4 et 5.

Étudier le systèmes d'exploitation UNIX ?

- 1 Utilisateur :
savoir utiliser le système UNIX.
- 2 Administrateur-programmeur :
savoir gérer un système UNIX : programmation shell-script.
- 3 Programmeur :
savoir bien utiliser les appels système (interface POSIX) dans un programme.
- 4 Théorie :
les problèmes posés par la conception d'un système d'exploitation.
- 5 Développeur de systèmes:
problèmes posé par l'implantation d'un système d'exploitation.

Ici: on s'occupera de 3,4 et 5.

Plan

- 1 Introduction
 - Que est-ce qu'un système d'exploitation
 - Un peu d'histoire
 - Introduction à Unix
- 2 Introduction aux processus
 - Généralités
 - États d'un processus
- 3 Éléments d'ordonnancement
 - Généralités
 - Ordonnancement avec réquisition (systèmes temps partagée)

Processus :

copie en mémoire d'un fichier exécutable,
... en train de s'exécuter.

Processus :

copie en mémoire d'un fichier exécutable,
... en train de s'exécuter.

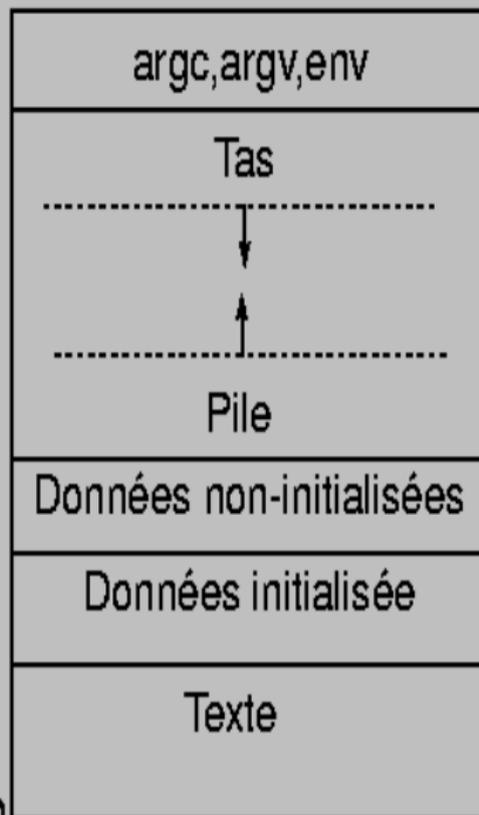
Un fichier exécutable ...

- en tête,
- section TEXT : le code,
- section BSS : données non allouées,
- section DATA : données initialisés,
- autres sections : symboles pour le déboguer, images,

...et la commande size

```
[lsantoca@localhost solutions2]$ make ex1; ls -l ex1; size -A ex1
gcc  identique.o  -o identique
mv identique ex1
-rwxr-xr-x    1 lsantoca lsantoca   20770 oct 23 14:27 ex1*
ex1 :
section          size          addr
...
.text            884      134513632
.data            12      134519016
.bss             20      134519296
...
.debug_info     7574           0
...
Total           15705
```

Adresse Haute = 0xFFFFFFFF



} initialisée à zéro
par exec
lu par exec

Adresse Basse = 0

Processus :

copie en mémoire d'un fichier exécutable,
... en train de s'exécuter.

Système multitache :

plusieurs processus en train de s'exécuter
au même temps sur une seule CPU.

Processus :

copie en mémoire d'un fichier exécutable,
... en train de s'exécuter.

Système multitache :

plusieurs processus en train de s'exécuter
au même temps sur une seule CPU.

Processus :

copie en mémoire d'un fichier exécutable,
... en train de s'exécuter.

Système multitache :

plusieurs processus en train de s'exécuter
au même temps sur une seule CPU.

Protection des autres processus :

partage de la mémoire :

chaque processus a son espace d'adressage privé.

2 modes d'exécution matérielle :

- mode utilisateur :
accès aux données privées du processus.
- mode superviseur, mode noyau :
accès aux données globales du système.

Passage entre les modes :

- appel système (trap) demandé par le processus,
- interruption matérielle.

Protection des autres processus :

partage de la mémoire :

chaque processus a son espace d'adressage privé.

2 modes d'exécution matérielle :

- mode utilisateur :
accès aux données privées du processus.
- mode superviseur, mode noyau :
accès aux données globales du système.

Passage entre les modes :

- appel système (trap) demandé par le processus,
- interruption matérielle.

Protection des autres processus :

partage de la mémoire :
chaque processus a son espace d'adressage privé.

2 modes d'exécution matérielle :

- mode utilisateur :
accès aux données privées du processus.
- mode superviseur, mode noyau :
accès aux données globales du système.

Passage entre les modes :

- appel système (trap) demandé par le processus,
- interruption matérielle.

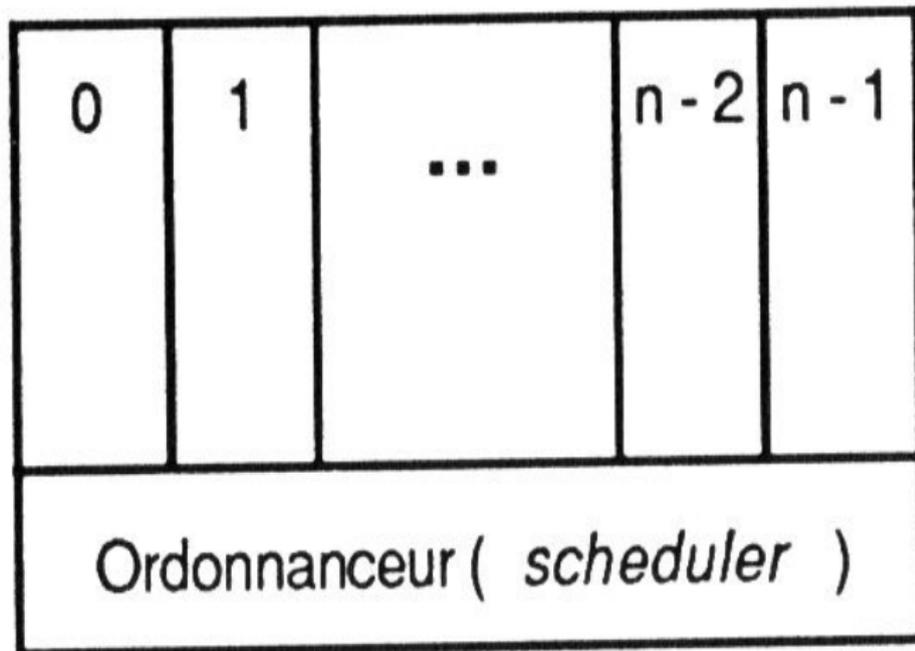
« plusieurs processus en train de s'exécuter ...

... au même temps sur une seule CPU »

partage du temps de calcul :

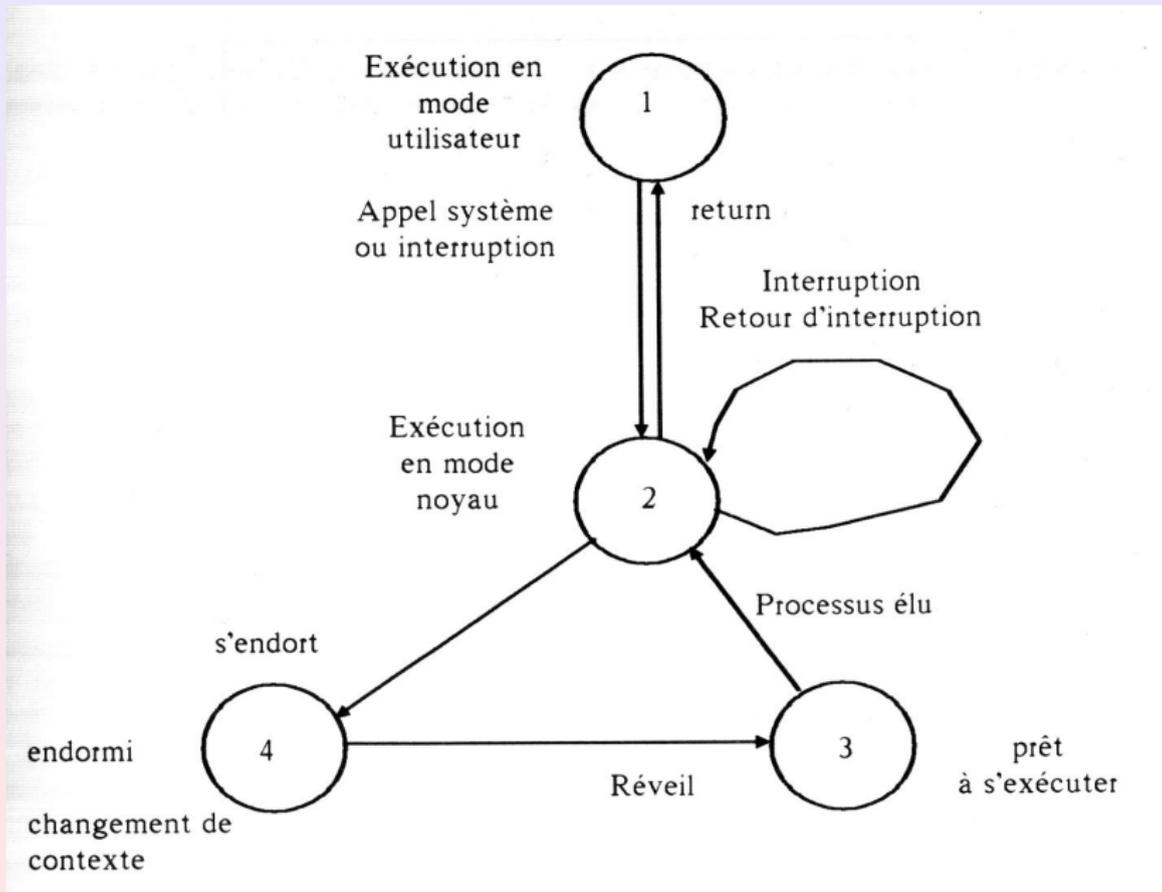
le noyau est (entra autre) *ordonnanceur*,
il s'occupe de repartir le temps mémoire.

Processus



Plan

- 1 Introduction
 - Que est-ce qu'un système d'exploitation
 - Un peu d'histoire
 - Introduction à Unix
- 2 Introduction aux processus
 - Généralités
 - États d'un processus
- 3 Éléments d'ordonnancement
 - Généralités
 - Ordonnancement avec réquisition (systèmes temps partagée)



Plan

- 1 Introduction
 - Que est-ce qu'un système d'exploitation
 - Un peu d'histoire
 - Introduction à Unix
- 2 Introduction aux processus
 - Généralités
 - États d'un processus
- 3 Éléments d'ordonnancement
 - Généralités
 - Ordonnancement avec réquisition (systèmes temps partagée)

Objectifs

Optimiser les paramètres suivants :

- taux d'utilisation de la CPU,
- débit,

- temps de traitement moyen,
- temps de traitement total,
- temps de réponse maximum.

Objectifs

Optimiser les paramètres suivants :

- taux d'utilisation de la CPU,
- débit,

- temps de traitement moyen,
- temps de traitement total,
- temps de réponse maximum.

Ordonnancement sans réquisition

- Algorithme FIFO.

	τ_i	t_i
P_1	5	0
P_2	6	0
P_3	1	6

Ordonnancement sans réquisition

- Algorithme FIFO.
- Algorithme PCTE
(plus court temps d'exécution) :
 - optimise le temps de traitement moyen,
 - on ne peut pas connaître le temps d'exécution,
il s'agit d'une mesure de comparaison.

	τ_i	t_i
P_1	5	0
P_2	6	0
P_3	1	6

Plan

- 1 Introduction
 - Que est-ce qu'un système d'exploitation
 - Un peu d'histoire
 - Introduction à Unix
- 2 Introduction aux processus
 - Généralités
 - États d'un processus
- 3 Éléments d'ordonnancement
 - Généralités
 - Ordonnancement avec réquisition (systèmes temps partagée)

Ordonnancement circulaire (tourniquet).

- On choisit le processus premier dans la file d'attente.
- Le processus est exécuté pendant au plus un *quantum* de temps.
- Si, il n'est pas terminé après le quantum, il est ajouté à la queue des la file d'attente.

Paramètre : le *quantum* du temps.

Choix de la durée du quantum :

Quantum bref : overhead.

Quantum long : temps de réponse important.

Ordonnancement circulaire (tourniquet).

- On choisit le processus premier dans la file d'attente.
- Le processus est exécuté pendant au plus un *quantum* de temps.
- Si, il n'est pas terminé après le quantum, il est ajouté à la queue des la file d'attente.

Paramètre : le *quantum* du temps.

Choix de la durée du quantum :

Quantum bref : overhead.

Quantum long : temps de réponse important.

Ordonnancement circulaire (tourniquet).

- On choisit le processus premier dans la file d'attente.
- Le processus est exécuté pendant au plus un *quantum* de temps.
- Si, il n'est pas terminé après le quantum, il est ajouté à la queue des la file d'attente.

Paramètre : le *quantum* du temps.

Choix de la durée du quantum :

Quantum bref : overhead.

Quantum long : temps de réponse important.

Ordonnancement circulaire (tourniquet).

- On choisit le processus premier dans la file d'attente.
- Le processus est exécuté pendant au plus un *quantum* de temps.
- Si, il n'est pas terminé après le quantum, il est ajouté à la queue des la file d'attente.

Paramètre : le *quantum* du temps.

Choix de la durée du quantum :

Quantum bref : overhead.

Quantum long : temps de réponse important.

Ordonnancement PCTER

PCTER : Plus Court Temps d'Exécution Restant.

- On choisit le processus dans la file d'attente selon le plus court TER.
- Le processus est exécuté au plus pendant un quantum de temps.
- Si, après ce quantum, il n'est pas terminé, il est ajouté à la file d'attente, et son nouveaux TER' est calculé selon la formule :

$$\text{TER}' = \text{TER} - \text{quantum}$$

Paramètre : le *quantum* du temps.

Analogie du PCTE, algorithme théorique car les temps d'Exécution n'est pas en général connu.

Ordonnancement PCTER

PCTER : Plus Court Temps d'Exécution Restant.

- On choisit le processus dans la file d'attente selon le plus court TER.
- Le processus est exécuté au plus pendant un quantum de temps.
- Si, après ce quantum, il n'est pas terminé, il est ajouté à la file d'attente, et son nouveaux TER' est calculé selon la formule :

$$\text{TER}' = \text{TER} - \text{quantum}$$

Paramètre : le *quantum* du temps.

Analogie du PCTE, algorithme théorique car les temps d'Exécution n'est pas en général connu.

Ordonnancement PCTER

PCTER : Plus Court Temps d'Exécution Restant.

- On choisit le processus dans la file d'attente selon le plus court TER.
- Le processus est exécuté au plus pendant un quantum de temps.
- Si, après ce quantum, il n'est pas terminé, il est ajouté à la file d'attente, et son nouveaux TER' est calculé selon la formule :

$$\text{TER}' = \text{TER} - \text{quantum}$$

Paramètre : le *quantum* du temps.

Analogue du PCTE, algorithme théorique car les temps d'Exécution n'est pas en général connu.

Groupes de processus

Deux types de processus :

- utilisation importante de la CPU,
- processus interactifs, utilisation importante des E/S.

But : favoriser les processus interactifs.

Idée : grouper les processus selon une priorité donnée.

Groupes de processus

Deux types de processus :

- utilisation importante de la CPU,
- processus interactifs, utilisation importante des E/S.

But : favoriser les processus interactifs.

Idée : grouper les processus selon une priorité donnée.

Groupes de processus

Deux types de processus :

- utilisation importante de la CPU,
- processus interactifs, utilisation importante des E/S.

But : favoriser les processus interactifs.

Idée : grouper les processus selon une priorité donnée.

Ordonnancement avec priorités.

- Chaque processus a une priorité de base, selon son type.
- Le processus dans la file d'attente avec la priorité plus basse est choisi.
- Si, après le quantum, le processus n'est pas terminé, il est préempté et ajouté dans la file d'attente.
- Le ré-calcul de la priorité se fait chaque délai selon la formule :
$$\text{priorité} = (\text{utilisation cpu})/2 + (\text{priorité de base}) .$$

Paramètres : quantum, priorité, délai.

Dans le UNIX System V, le délai vaut 1 seconde.

```
algorithme schedule_process
```

```
entrée: néant
```

```
sortie: néant
```

```
{
```

```
  while (aucun processus à prélever en vue de son exécution)
```

```
  {
```

```
    for ( chaque processus dans la file d'attente d'exécution )
```

```
      prélever le processus de plus haute priorité qui est
```

```
      chargé en mémoire;
```

```
    if (aucun processus éligible en vue de son exécution)
```

```
      rendre la machine oisive;
```

```
    /* une interruption sortira la machine de son état oisif */
```

```
  }
```

```
  extraire le processus choisi de la file d'attente d'exécution;
```

```
  changer de contexte pour le processus élu,
```

```
  reprendre son exécution;
```

```
}
```