



Titre: Les Systèmes d'Exploitation

Auteurs: Bertrand Collin, Marc Baudoin, Manuel Bouyer, Jérôme Gueydan, Thomas Degris, Frédéric Loyer, Damien Mercier.

Ecole/Université: [École nationale supérieure de techniques avancées](#)

Résumé: Pourquoi un système d'exploitation ? Le système d'exploitation est l'élément essentiel qui relie la machine, composée d'éléments physiques comme le microprocesseur, le disque dur ou les barrettes mémoire, et l'utilisateur qui souhaite effectuer des calculs. Sans système d'exploitation, chaque utilisateur serait obligé de connaître le fonctionnement exact de la machine sur laquelle il travaille et il serait, par exemple, amené à programmer directement un contrôleur de périphérique USB pour pouvoir enregistrer ses [données](#) sur une clé USB.

Sans le contrôle du système d'exploitation, les utilisateurs pourraient aussi détruire irrémédiablement certaines données stockées sur un ordinateur, voire détruire certains périphériques - comme le disque dur - en programmant des opérations illicites. Les systèmes d'exploitation jouent donc un rôle majeur dans l'utilisation d'un ordinateur et si l'outil informatique s'est répandu dans le monde entier, c'est certes grâce à l'abaissement des prix d'achat et grâce à l'augmentation de la puissance des ordinateurs, mais c'est surtout grâce aux progrès réalisés lors des cinquante dernières années dans la programmation des systèmes d'exploitation : avec une machine de puissance équivalente, la moindre opération sur un ordinateur qui aujourd'hui nous paraît triviale était alors proprement impossible à réaliser !

Ce document présente les systèmes d'exploitation ainsi que leurs composants principaux et leurs structures. Toutes les notions essentielles à la compréhension du fonctionnement d'un ordinateur avec son système d'exploitation sont ici abordées, mais un lecteur désireux d'approfondir un sujet particulier doit se reporter à un ouvrage spécialisé, par exemple un des ouvrages cités dans la bibliographie. Les notions présentées ici ne font en général référence à aucun système d'exploitation en particulier. Néanmoins, nous avons choisi d'illustrer les fonctionnalités avancées des systèmes d'exploitation par des exemples provenant essentiellement des systèmes d'exploitation de type Unix. Nous justifierons plus



loin ce choix et un lecteur critique pourra considérer que ce choix est a priori aussi valable qu'un autre. Nous abordons aussi dans ce document certains sujets qui ne relèvent pas directement des systèmes d'exploitation, comme l'architecture des ordinateurs ou la compilation de programmes, mais dont l'étude permet d'expliquer plus facilement le rôle d'un système d'exploitation et de montrer l'étendue des services qu'il rend.

Pourquoi étudier les systèmes d'exploitation ? Avant de se lancer à corps perdu dans l'étude des systèmes d'exploitation, il est raisonnable de se demander ce que cette étude peut nous apporter. Précisons tout d'abord que ce document est à l'origine le support écrit d'un **cours** proposé en deuxième année du cycle d'ingénieur de l'ENSTA ParisTech 2. Les raisons évoquées ci-dessous s'adressent donc à tous les étudiants non spécialisés en informatique qui exerceront rapidement des métiers d'ingénieur à responsabilité. Tous les ingénieurs issus de grandes écoles généralistes auront à utiliser l'outil informatique dans leur métier. Bien entendu, suivant le métier exercé ou suivant l'évolution de la carrière de chacun, cette utilisation sera plus ou moins fréquente et certains ne feront que tapoter de temps en temps sur leur **clavier** alors que d'autres passeront des heures à se battre avec (contre ?) la machine. Quel que soit le profil choisi, tous les ingénieurs de l'ENSTA ParisTech se retrouveront dans une des trois 3 catégories suivantes.

L'utilisateur : comme son nom l'indique, la seule préoccupation de l'utilisateur est d'utiliser sa machine. Son désir le plus vif est que celle-ci se mette à fonctionner normalement quand il l'allume et que ses **logiciels** favoris lui permettent de travailler correctement. Le décideur : il prend les décisions vitales concernant les choix stratégiques et commerciaux de l'informatique d'entreprise. C'est lui qui décidera par exemple s'il vaut mieux acheter un gros ordinateur relié à 50 terminaux ou s'il est préférable d'acheter 50 micro-ordinateurs.

Nous pourrions ajouter d'autres catégories afin de distinguer certaines professions comme par exemple le conseil en informatique connectés en réseau. Souvent le décideur se fonde sur les besoins exprimés par les utilisateurs pour prendre sa décision. Le programmeur : il cherche à tirer le meilleur parti de la machine qu'il programme tout en perdant le moins de temps possible en développements. Le programmeur cherche aussi à préserver ses programmes du temps qui passe et tente de les réutiliser de machine en machine afin de ne pas tout recommencer à chaque fois. Beaucoup d'élèves ingénieurs estiment spontanément



que seule la 3e catégorie doit s'intéresser aux cours d'informatique et, notamment, aux systèmes d'exploitation. En fait, ceci est faux et c'est même une erreur grave.

Extrait du sommaire:

Glossaire 9

Introduction 11

Pourquoi un système d'exploitation ? 11

Pourquoi étudier les systèmes d'exploitation ? 12

Plan du document 16

Les sujets qui ne sont pas traités dans ce document 18

Table des matières 19

I Les systèmes d'exploitation 25

1 Rappels sur l'architecture des ordinateurs 27

1.1 Représentation symbolique et minimaliste d'un ordinateur 27

1.2 Représentation fonctionnelle d'un ordinateur 28

1.3 Mode noyau versus mode utilisateur 40

1.4 Le jeu d'instructions 41

1.5 Rôle de l'unité de gestion de mémoire 45

1.6 Performances des ordinateurs 47

1.7 Conclusion : que retenir de ce chapitre ? 53

2 Qu'est-ce qu'un système d'exploitation ? 55

2.1 Définitions et conséquences 56

2.2 Les appels système 58

2.3 Structure d'un système d'exploitation 60

2.4 Les différents types de systèmes d'exploitation 64

2.5 Les services des systèmes d'exploitation 66

2.6 Conclusion : que retenir de ce chapitre ? 67

3 Évolution des systèmes d'exploitation 69

3.1 Les origines et les mythes (16xx-1940) 70

3.2 La préhistoire (1940-1955) 70

3.3 Les ordinateurs à [transistor](#) (1955-1965) 74



- 3.4 Les circuits intégrés (1965-1980) 79
- 3.5 L'informatique moderne (1980-1995) 85
- 3.6 Les systèmes d'exploitation d'aujourd'hui 86
- 3.7 Conclusion : que faut-il retenir de ce chapitre ? 96
- 4 Compilation et édition de liens 97
 - 4.1 Vocabulaire et définitions 100
 - 4.2 Les phases de compilation 104
 - 4.3 L'édition de liens 111
 - 4.4 Structure des fichiers produits par compilation 115
 - 4.5 Les problèmes d'adresses 117
 - 4.6 Les appels système 119
- 5 La gestion des processus 121
 - 5.1 Qu'est-ce qu'un processus ? 122
 - 5.2 La hiérarchie des processus 126
 - 5.3 Structures utilisées pour la gestion des processus 132
 - 5.4 L'ordonnancement des processus (scheduling) 139
 - 5.5 Conclusion 149
- 6 La gestion de la mémoire 151
 - 6.1 Les adresses virtuelles et les adresses physiques 152
 - 6.2 Les services du gestionnaire de mémoire 157
 - 6.3 La gestion de la mémoire sans pagination 162
 - 6.4 La pagination 165
 - 6.5 Conclusion 168
- 7 Le système de fichiers 169
 - 7.1 Les services d'un système de fichier 170
 - 7.2 L'organisation des systèmes de fichiers 176
 - 7.3 La gestion des volumes 182
 - 7.4 Améliorations des systèmes de fichiers 185
- 8 Les architectures multi-processeurs et les threads 191
 - 8.1 De la loi de Moore au multi-coeurs 191
 - 8.2 Les threads 199



- 8.3 Conclusion 210
- 9 La virtualisation des systèmes d'exploitation 211
 - 9.1 Les intérêts et les enjeux de la virtualisation 212
 - 9.2 Les différentes solutions 216
 - 9.3 Conclusion 228
- II Programmation système en C sous Unix 229
- 10 Les processus sous Unix 231
 - 10.1 Introduction 231
 - 10.2 Les entrées / sorties en ligne 231
 - 10.3 Les **fonctions** d'identification des processus 235
 - 10.4 Exercices 236
 - 10.5 Corrigés 238
 - 10.6 Corrections détaillées 239
- 11 Les entrées / sorties sous Unix 245
 - 11.1 Les descripteurs de fichiers 246
 - 11.2 Les appels système associés aux descripteurs de fichier 246
 - 11.3 La bibliothèque standard d'entrées / sorties 253
 - 11.4 Exercices 258
 - 11.5 Corrigés 259
 - 11.6 Corrections détaillées 261
- 12 Création de processus sous Unix 267
 - 12.1 La création de processus 267
 - 12.2 L'appel système wait() 268
 - 12.3 Exercices 269
 - 12.4 Corrigés 270
 - 12.5 Corrections détaillées 273
- 13 Recouvrement de processus sous Unix 289
 - 13.1 Les appels système de recouvrement de processus 289
 - 13.2 Exercices 291
 - 13.3 Corrigés 293
- 14 Manipulation des signaux sous Unix 297



- 14.1 Liste et signification des différents signaux 297
- 14.2 Envoi d'un signal 298
- 14.3 Interface de programmation 299
- 14.4 Conclusion 302
- 14.5 Exercices 303
- 14.6 Corrigés 304
- 15 Les tuyaux sous Unix 307
 - 15.1 Manipulation des tuyaux 308
 - 15.2 Exercices 318
 - 15.3 Corrigés 320
 - 15.4 Corrections détaillées 329
- 16 Les sockets sous Unix 347
 - 16.1 Introduction 347
 - 16.2 Les RFC request for comments 347
 - 16.3 Technologies de communication réseau 347
 - 16.4 Le protocole IP 348
 - 16.5 Interface de programmation 351
- 17 Les threads POSIX sous Unix 363
 - 17.1 Présentation 363
 - 17.2 Exercices 370
 - 17.3 Corrigés 371
 - 17.4 Architectures et programmation parallèle 379
- 18 Surveillances des entrées / sorties sous Unix 381
 - 18.1 Introduction 381
 - 18.2 L'appel système select() 382
 - 18.3 Exercices 383
 - 18.4 Corrigés 385
- 19 Utilisation d'un débogueur 393
 - 19.1 Introduction 393
 - 19.2 gdb, xgdb, ddd et les autres 394
 - 19.3 Utilisation de gdb 394



- III Projets 407
 - 20 Projet de carte bleue 409
 - 20.1 Introduction 409
 - 20.2 Cahier des charges 411
 - 20.3 Conduite du projet 416
 - 20.4 Évolutions complémentaires et optionnelles 420
 - 20.5 Annexes 421
 - 21 Le Scaphandre et le Papillon 427
 - 21.1 Énoncé 427
 - 21.2 Contraintes générales 427
 - 21.3 Contraintes techniques 428
 - 21.4 Services à mettre en oeuvre 429
 - 21.5 Précisions 431
 - 21.6 Déroulement du projet 431
 - 22 Projet de Jukebox 433
 - 22.1 Énoncé 433
 - 22.2 Outils utilisés 433
 - 22.3 Commandes de mpg123 434
 - 22.4 Les ID3-tag 437
 - 22.5 Suggestion d'un plan d'action 437
 - 23 Administration distante 439
 - 23.1 Position du problème 439
 - 23.2 Le processus Machine 440
 - 23.3 Le processus Administrateur 441
 - 23.4 L'interface 444
 - 23.5 Les dialogues 444
 - 23.6 Consignes et cahier des charges 445
 - 23.7 Améliorations possibles 447
 - 23.8 Aide mémoire 448
 - 24 Parcours multi-agents dans un [labyrinthe](#) 455
 - 24.1 Position du problème 456



- 24.2 L'organisation générale 461
- 24.3 Mise en oeuvre du projet 465
- 24.4 Conseils en vrac 469
- 25 Course de voitures multi-threads 471
 - 25.1 Présentation 471
 - 25.2 Cahier des charges 476
 - 25.3 Annexes 479
- IV L'aide-mémoire du langage C 487
- 26 Aide-mémoire de langage C 489
 - 26.1 Éléments de syntaxe 491
 - 26.2 La compilation sous Unix 493
 - 26.3 Types, opérateurs, expressions 500
 - 26.4 Tests et branchements 514
 - 26.5 Tableaux et [pointeurs](#) 518
 - 26.6 Le préprocesseur 527
 - 26.7 Fonctions d'entrée/sortie 531
 - 26.8 Compléments : la bibliothèque standard et quelques fonctions annexes 534
- Index - Général 553
- Index - Programmation 557
- Bibliographie 559

[Cours divers en électronique 6](#)

Télécharger le fichier PDF: [Les Systèmes d'Exploitation](#)