Références sciences

Unix & Linux

Utilisation et administration
avec plus de 400 exercices corrigés

Jean-Michel Léry

ellipses

9782340-038004_001_828.indd 1 01/04/2020 16:34

Collection Références sciences

dirigée par Paul de Laboulaye
paul.delaboulaye@editions-ellipses.fr

Retrouvez tous les livres de la collection et des extraits sur www.editions-ellipses.fr

ISBN 9782340-038004
©Ellipses Édition Marketing S.A., 2020
32, rue Bargue 75740 Paris cedex 15



Le Code de la propriété intellectuelle n'autorisant, aux termes de l'article L. 122-5.2° et 3°a), d'une part, que les « copies ou reproductions strictement réservées à l'usage privé du copiste et non destinées à une utilisation collective », et d'autre part, que les analyses et les courtes citations dans un but d'exemple et d'illustration, « toute représentation ou reproduction intégrale ou partielle faite sans le consentement de l'auteur ou de ses ayants droit ou ayants cause est illicite » (art. L. 122-4).
Cette représentation ou reproduction, par quelque procédé que ce soit constituerait une contrefaçon sanctionnée par les articles L. 335-2 et suivants du Code de la propriété intellectuelle.

www.editions-ellipses.fr

9782340-038004_001_828.indd 2 01/04/2020 16:34

Table des matières

Introduction **1**

PARTIE I – Utilisation **7**

1. Présentation du système UNIX **9**

1. Bref historique d'UNIX **9**

1.1. Histoire d'UNIX 9
1.2. UNIX et les logiciels libres 11
1.3. Histoire de Linux 11
1.4. Les versions du noyau de Linux 13
1.5. Les distributions Linux 13

2. Les caractéristiques d'UNIX **15**

2.1. Un système multiplateforme 15
2.2. Un système multitâche et multi-utilisateur 15
2.3. Une hiérarchie arborescente gérée en « système de fichiers » 16
2.4. Différents environnements de commandes : les shells 16
2.5. Gestion hiérarchisée des processus 17
2.6. Un système ouvert 17
2.7. Différents UNIX : AIX (IBM), HP-UX (HP), Solaris (Oracle), Linux 18
2.8. Un lien fort avec le langage C 18

Résumé **18**

Questions de révisions **19**

Exercices **19**

2. L'accès au système **21**

1. La connexion directe du poste de travail **21**

1.1. Principe 21
1.2. La notion de terminal 22
1.3. L'émulation de terminal 23
1.4. Le terminal sous Linux : la console 24

2. L'accès par le réseau **25**

2.1. Principe 25
2.2. Adresses IP et DNS 26
2.3. Services réseaux et modes d'accès 29

Résumé **32**

Questions de révisions **32**

Exercices **32**

3. La première session **33**

1. La connexion **33**

1.1. Les interfaces 34

1.2. L'authentification 38
1.3. Le changement du mot de passe 39
2. La déconnexion 40
2.1. En mode graphique 40
2.2. En mode commande 41
3. L'environnement standard de travail 41
3.1. Le répertoire personnel ou *homedir* 42
3.2. Le login shell 42
3.3. L'invite de commandes, ou *prompt* 43
3.4. Le positionnement du type de terminal 43
4. Les utilisateurs connectés 44
4.1. La commande *who* 44
4.2. La commande *w* 45
4.3. La commande *finger* sans argument 46
4.4. Le filtrage de l'affichage 47
5. L'identification d'un utilisateur 49
5.1. Principe 49
5.2. Les commandes d'identification 50
5.3. Le fichier */etc/passwd* 52
5.4. Le fichier */etc/group* 54
5.5. Les modifications des informations d'un utilisateur 54
Résumé 57
Questions de révisions 57
Exercices 57

4. Les fichiers et les répertoires 59
1. Notions sur les fichiers et les répertoires 59
1.1. Principe 59
1.2. Concept de fichier 60
1.3. Concept de répertoire 61
1.4. Nom d'un fichier ou d'un répertoire 62
2. La hiérarchie 68
2.1. Présentation de l'arborescence 69
2.2. Les espaces de travail 71
2.3. Les chemins 72
3. Les commandes de base 77
3.1. Syntaxe d'une ligne de commandes 77
3.2. L'ensemble des commandes 79
3.3. La gestion des fichiers 80
3.4. La gestion des répertoires 84
3.5. Exemple de session 93
3.6. Notion de *builtin* et de « commande externe » 95
Résumé 96
Questions de révisions 97
Exercices 97

5. Gestion des fichiers sous GNOME 99
1. Le logiciel GNOME Fichiers 99
1.1. Lancement 99

1.2. Description de l'interface 100
1.3. Les menus contextuels 101
1.4. Configuration de GNOME Fichiers 102
2. Afficher 103
2.1. Afficher les fichiers cachés 104
2.2. Afficher les propriétés d'un fichier 104
2.3. Afficher le contenu d'un fichier 105
2.4. Afficher en liste 106
3. Copier, renommer, déplacer 107
3.1. Copier un fichier 107
3.2. Renommer un fichier 107
3.3. Déplacer un fichier 108
4. Créer et détruire 109
4.1. Créer un répertoire 109
4.2. Mettre un fichier à la corbeille 109
5. Se déplacer dans un répertoire 110
6. Chercher un fichier ou un répertoire 110
Résumé 111
Questions de révisions 112
Exercices 112

6. La gestion de l'espace disque 113

1. Les unités de mesure de l'espace disque 113
1.1. Le choix historique 113
1.2. La nouvelle nomenclature 114
2. Commandes avancées 116
2.1. La commande *file* 116
2.2. La commande *touch* 117
2.3. La commande *find* 118
2.4. La commande *diff* 125
2.5. La commande *du* 126
3. Accès à la disquette, à la clef USB et au CD-ROM 126
3.1. La gestion des périphériques 126
3.2. Le mode de commande UNIX 128
3.3. L'interface graphique de GNOME 130
3.4. Les commandes MS-DOS émulées sous Linux 131
4. La partition Windows 134
4.1. Principe de « l'amorçage multiple » ou « multiboot » 134
4.2. Accès aux partitions Windows 134
5. Notions sur les systèmes de fichiers 136
5.1. Principe 136
5.2. Montage et démontage, *mount* et *umount* 138
5.3. La commande *df* 140
6. Les quotas de disque 141
6.1. Principe 141
6.2. Les commandes 142
7. Les liens 143
7.1. Les liens symboliques 143
7.2. Les liens physiques 144

Résumé **144**
Questions de révisions **145**
Exercices **145**

7. Les droits **147**
1. Principe **148**
2. Rappel sur la notion d'utilisateur et de groupe **148**
3. Les attributs d'un fichier **150**
3.1. Le propriétaire et le groupe propriétaire 150
3.2. Les droits 151
4. La gestion des droits **152**
4.1. L'affichage 152
4.2. La modification : la commande *chmod* 154
4.3. Modification des droits sous GNOME 161
4.4. Les grandes familles de droits 161
5. L'interprétation des droits **162**
5.1. Pour un fichier 162
5.2. Pour un répertoire 166
6. Le changement de propriétaire : la commande *chown* **171**
7. Le changement de groupe : la commande *chgrp* **174**
8. Les droits par défaut : la commande *umask* **175**
8.1. Principe 175
8.2. Le masque octal 176
8.3. Exemple de session 177
8.4. Les grandes familles de masques 179
9. La gestion du groupe actif : la commande *newgrp* **179**
10. Les droits d'endossement **180**
10.1. Le SUID-bit 181
10.2. Le SGID-bit 185
11. Le sticky-bit **187**
12. Les ACL AIX **188**
12.1. Format des ACL 189
12.2. Affichage 189
12.3. Modification 190
Résumé **191**
Questions de révisions **192**
Exercices **193**

8. L'environnement de travail ou shell **195**
1. Principe **195**
2. Les shells sous UNIX **196**
2.1. Les shells standard 196
2.2. Les autres shells 197
2.3. Le shell POSIX 197
2.4. Changement de shell 198
2.5. La documentation sur les shells 199
2.6. Choix d'un environnement 199
2.7. Caractéristiques des shells 200

3. Les variables d'environnement 205
3.1. Principe 205
3.2. Gestion des variables d'environnement 206
3.3. Portée des variables 214
3.4. Les principales variables 218
4. Les alias 229
4.1. Affichage 229
4.2. Modification 230
4.3. Suppression 231
5. Les fichiers de démarrage 232
5.1. Principe 232
5.2. À l'ouverture d'une session 232
5.3. À l'exécution d'un shell 236
5.4. Pour les logiciels applicatifs 237
5.5. Le fichier */etc/environment* 237
6. Le système d'historique des commandes 238
6.1. Positionnement de l'historique (paramètres) 239
6.2. Utilisation de l'historique 241
7. La complétion des commandes 243
8. Le changement de session : *su* et *login* 244
9. L'environnement du bureau GNOME 245
9.1. L'outil *gnome-control-center* 245
9.2. L'outil *gnome-tweak-tool* 247
Résumé 248
Questions de révisions 249
Exercices 249

9. Les processus 251

1. Principe 251
2. La hiérarchie des processus 252
3. La communication interprocessus, les signaux 253
4. La gestion des processus interactifs 255
5. La gestion de tous les processus 257
5.1. Les commandes 258
5.2. Affichage des processus : la commande *ps* 258
5.3. Envoi de signaux : la commande *kill* 261
5.4. Gestion des processus liés aux répertoires : la commande *fuser* 264
5.5. Changement des priorités : les commandes *nice* et *renice* 265
6. Les traitements différés 266
6.1. La commande *nohup* 266
6.2. La commande *at* 267
6.3. La commande *crontab* 270
7. Les démons 272
Résumé 274
Questions de révisions 275
Exercices 276

10. Le réseau 277

1. Principe 277
2. Les commandes réseau 278
2.1. Les commandes « outils » 278
2.2. Le transfert de fichiers par *ftp* 282
2.3. Les « remote » commandes 287
2.4. Les connexions et les échanges sécurisés 292
3. Les outils de communication 296
3.1. La communication synchrone 297
3.2. Le courrier électronique 300
Résumé 308
Questions de révisions 309
Exercices 310

11. Compléments 311
1. Principe 311
2. L'éditeur de texte *vi* 312
2.1. Les modes de fonctionnement 312
2.2. Le mode commande 314
2.3. Le mode EX 317
2.4. Paramétrage et fichier de démarrage 320
2.5. Les variantes de *vi* 321
2.6. Exemple de session 322
3. L'impression 323
4. La sauvegarde et l'archivage 325
4.1. La commande *tar* 325
4.2. Les commandes AIX *backup* et *restore* 327
4.3. Les commandes Linux *dump* et *restore* 328
4.4. La commande *dd* 330
5. Les outils de compression 331
5.1. Les commandes *compress* et *uncompress* 331
5.2. Les commandes *gzip* et *gunzip* 332
6. L'aide en ligne 332
6.1. La commande *man* 333
6.2. L'option --*help* sous Linux 336
6.3. La documentation sous GNOME 336
6.4. Les HOWTO et les FAQ 337
7. La gestion des entrées/sorties et les filtres 338
7.1. Les canaux de communication 338
7.2. Redirection des entrées/sorties 339
7.3. Les tubes ou « pipes » 341
7.4. Les commandes de filtrage 341
Résumé 345
Questions de révisions 346
Exercices 346

PARTIE II – Administration 349

12. L'installation de Linux 351
1. Principe 351

2. Notions sur le partitionnement 352
2.1. Pourquoi partitionner ? 352
2.2. Les limitations du partitionnement 354
2.3. L'ordre des partitions 358
3. Préparation de l'installation 358
3.1. Faire l'inventaire de son matériel 358
3.2. Choisir entre poste de travail et serveur 360
3.3. Le plan de partitionnement 361
4. Créer ses DVD ou ses clefs USB d'installation 374
4.1. Télécharger les images Linux 375
4.2. Graver le DVD de démarrage 375
4.3. Créer une clef USB de démarrage 375
4.4. Lancer l'installation 380
5. Les étapes de l'installation 381
5.1. Installation du poste de travail 382
5.2. Installation du serveur 393
6. Installation des logiciels sous Linux 398
6.1. Les *paquets* logiciels 398
6.2. Les gestionnaires de paquets sous Ubuntu 398
6.3. Les gestionnaires de paquets RPM 410
6.4. Les fichiers *tar* compressés 421
Résumé 425
Questions de révisions 426
Exercices 426

13. La configuration du système 427
1. Principe 427
2. Les outils d'administration 428
2.1. Les commandes 428
2.2. Les interfaces propriétaires 428
2.3. Les interfaces Linux 430
3. Le réseau 438
3.1. Le système d'adressage 439
3.2. Le routage 448
3.3. Configuration de l'interface du serveur 456
3.4. Résolution des noms 460
3.5. Les services réseau 466
4. Le terminal 479
4.1. L'écran graphique 479
4.2. Définir un nouveau terminal 480
5. L'impression 483
5.1. Principe de fonctionnement 483
5.2. Arrêt et démarrage du service d'impression 483
5.3. Configurer une nouvelle imprimante 484
5.4. Gérer les files d'impression 492
5.5. Configurer le service d'impression 493
Résumé 500
Questions de révisions 501

Exercices **501**

14. L'exploitation du système 503

1. Principe **503**
2. Arrêt et démarrage **504**
2.1. Choix du logiciel de démarrage : LILO ou GRUB 504
2.2. Démarrage avec LILO 504
2.3. Démarrage avec GRUB 519
2.4. Démarrage d'un système UNIX 529
2.5. Arrêt d'un système UNIX 544
3. Gestion des services **545**
3.1. La méthode naïve 545
3.2. Sous Linux 546
3.3. Sous AIX 548
4. Mise en place de sauvegardes du système **550**
4.1. Sauvegarde de la hiérarchie 550
4.2. Sauvegarde de partitions du disque 558
5. Clonage de l'ordinateur et réinstallation **561**
5.1. À partir d'images des partitions du disque 561
5.2. Avec le logiciel « *gnome-disks* » 567
Résumé **572**
Questions de révisions **572**
Exercices **573**

15. La gestion des utilisateurs 575

1. Principe **575**
2. Administration des groupes **578**
2.1. Ajout d'un nouveau groupe 578
2.2. Suppression d'un groupe 581
3. Administration des utilisateurs **584**
3.1. Ajout d'un nouvel utilisateur 584
3.2. Suppression d'un utilisateur 591
4. Le « shadow password » **593**
4.1. Principe 594
4.2. Structure du fichier 594
4.3. Les commandes associées 596
5. Protection du compte *root* **598**
5.1. Protection sous Linux 598
5.2. Protection sous AIX 606
6. Gestion des mots de passe **607**
6.1. Copier-coller un mot de passe 607
6.2. Changer un mot de passe et verrouiller un compte 607
6.3. Changer la durée de vie ou le « password aging » 608
7. Les fichiers associés **610**
7.1. Le répertoire */etc/skel* 610
7.2. Le fichier */etc/login.defs* 610
7.3. Le répertoire */etc/security* 611
8. Automatisation de la création de comptes **612**

8.1. Principe de fonctionnement ... 612
8.2. Codage du mot de passe ... 613
8.3. Script d'automatisation ... 616
Résumé ... 618
Questions de révisions ... 618
Exercices ... 619

16. La gestion des systèmes de fichiers ... 621
1. Principe ... 621
2. Pourquoi créer plusieurs systèmes de fichiers ? ... 623
3. Présentation des systèmes de fichiers ... 624
3.1. Les différents types de systèmes de fichiers ... 624
3.2. Structure du système de fichiers *ext2* ... 625
3.3. Les systèmes de fichiers journalisés *ext3*, *ext4* et *jfs* ... 629
4. Création d'un système de fichiers ... 629
4.1. Partitionnement ... 630
4.2. La commande *mke2fs* ... 632
4.3. Montage manuel ... 634
4.4. Montage automatique ... 634
5. Gestion d'un système de fichiers ... 634
5.1. Affichage de la liste des systèmes de fichiers *df* et *du* ... 635
5.2. Montage et démontage ... 635
5.3. Vérification avec *fsck* ... 644
5.4. Restauration du super-bloc ... 645
5.5. Conversions entre les types *ext2*, *ext3* et *ext4* ... 646
5.6. Agrandir un système de fichiers ... 649
6. Les espaces de swap ... 651
6.1. Création d'une partition de swap ... 652
6.2. Création d'un fichier de swap ... 654
6.3. Désactivation d'un espace de swap ... 655
7. Le montage dynamique ... 655
7.1. Principe ... 655
7.2. Avec *systemd* ... 656
7.3. Avec *autofs* ... 667
8. La gestion des quotas disque ... 671
8.1. Principe ... 672
8.2. Installation du logiciel *quota* ... 673
8.3. Les commandes ... 673
8.4. Mise en œuvre ... 676
8.5. Utilisation ... 678
9. Les volumes logiques ... 679
9.1. Principe ... 680
9.2. Installation du *Logical Volume Manager* ... 681
9.3. Création des partitions ... 682
9.4. Mise en œuvre ... 683
9.5. Changement dynamique de la taille d'un système de fichiers ... 688
9.6. Ajout d'une nouvelle partition ... 689
9.7. Suppression d'un volume logique ... 691
9.8. Suppression d'un groupe de volumes ... 691

9.9. Suppression d'un volume physique 691
9.10. Modification des attributs d'un groupe de volumes 691
Résumé 692
Questions de révisions 693
Exercices 693

17. Les services réseau 695
1. Principe 695
2. Le service DNS 696
2.1. Principe de fonctionnement 696
2.2. Mise en œuvre 698
2.3. Configuration du poste client 711
3. Le service NFS 713
3.1. Principe de fonctionnement 714
3.2. Mise en œuvre 718
3.3. Configuration du poste client 720
4. Le service NIS 723
4.1. Installation du logiciel 724
4.2. Principe de fonctionnement 724
4.3. Les maps 725
4.4. Les problèmes de sécurité 726
4.5. Mise en œuvre des serveurs 727
4.6. Configuration du poste client 733
5. Le service de messagerie 737
5.1. Fonctionnement d'un service de messagerie 737
5.2. Le protocole SMTP 738
5.3. Sendmail 740
5.4. POP et IMAP 750
5.5. Configuration du poste client 756
Résumé 757
Questions de révisions 758
Exercices 759

Réponses et solutions 761

Index 789

Introduction

UNIX est un système d'exploitation très répandu. Il est omniprésent dans l'univers des serveurs et a fait une percée importante dans celui des micro-ordinateurs, avec les distributions Linux. Il est devenu incontournable pour tout administrateur système. Les ouvrages qui traitent de ce système portent sur son utilisation ou sur son administration, mais rares sont ceux qui abordent simultanément ces deux aspects. Ce livre a donc été conçu comme un « tout en un ». Il permet de débuter sous UNIX ou Linux, et de poursuivre son apprentissage jusqu'au niveau administrateur. L'intérêt de cette démarche est qu'elle facilite le passage du niveau utilisateur à celui d'administrateur, en présentant les concepts de façon homogène, et en approfondissant, dans la partie administration, les exemples étudiés dans la partie utilisation. Ainsi, dans la deuxième partie, le lecteur découvre comment mettre en œuvre les services qu'il a appris à utiliser dans la première partie du livre.

Nous proposons dans cet ouvrage une approche à la fois pédagogique et pratique des systèmes UNIX et Linux. Nous y exposons les concepts fondamentaux qui sont communs à l'ensemble des systèmes, mais aussi les particularités techniques de certaines versions. Nous nous appuyons sur des exemples et des exercices corrigés. Notre expérience de l'enseignement nous a montré que l'explication de concepts importants, parfois complexes à assimiler, doit toujours être accompagnée d'exemples pratiques et concrets pour asseoir les connaissances.

L'apprentissage du système UNIX pose le problème du choix de la distribution utilisée comme support aux exemples. Si la syntaxe des commandes utilisateurs est similaire d'un UNIX à l'autre, l'administration de systèmes repose parfois sur des commandes ou des interfaces graphiques propriétaires (c'est-à-dire particulières à un système d'exploitation). La finalité de cet ouvrage n'est pas de détailler les spécificités de chaque système, mais de présenter le système UNIX et ses concepts de manière globale et suffisamment claire pour permettre au lecteur de s'adapter à toute nouvelle distribution. Cependant, le choix d'une seule distribution pour les exemples ne permet pas de dissocier clairement le concept à assimiler de la technique de mise en œuvre qui lui est associée, et qui peut être particulière à la distribution. C'est pourquoi les exemples s'appuient principalement sur deux environnements système, Linux Ubuntu et AIX (IBM). Ce choix est motivé par l'importance de ces deux systèmes dans leur domaine respectif : le logiciel libre et

9782340-038004_001_828.indd 13 01/04/2020 16:34

les produits commerciaux. La partie « utilisation » de cet ouvrage propose des syntaxes communes aux deux environnements ; la partie « administration » privilégie le système Linux Ubuntu, tout en indiquant les variantes syntaxiques propres à AIX et parfois à Linux Fedora.

Public et progression pédagogique

Unix & Linux – Utilisation et administration s'adresse aussi bien aux débutants qu'aux utilisateurs avertis dans le domaine des systèmes d'exploitation. Il est cependant recommandé de posséder des connaissances générales en informatique et d'avoir déjà eu une expérience d'installation de poste de travail avec un autre système d'exploitation.

L'ouvrage est structuré en deux parties. La première est dédiée à l'apprentissage du système UNIX en tant qu'utilisateur, la seconde porte sur son administration. À la fin de la première partie, le lecteur doit avoir atteint le niveau « utilisateur averti ». Certains concepts d'administration y sont déjà abordés, de manière à préparer le second niveau. La première partie ne correspond donc pas à un simple apprentissage du niveau utilisateur : il s'agit également d'une préparation au niveau administrateur.

Chaque chapitre traite un sujet particulier et peut être abordé de manière autonome. Les objectifs à atteindre sont énoncés en début de chapitre, et les acquis sont vérifiés par des questions et des exercices à la fin.

La progression pédagogique est doublée d'une progression « pratique ». Ainsi, le chapitre d'installation d'une distribution Linux, qui est la première étape de l'administration d'un système, a été conçu pour permettre à un débutant d'installer son poste de travail (qui est nécessaire à l'apprentissage du niveau utilisateur). De la sorte, le lecteur peut installer « simplement » son ordinateur dans un premier temps, puis revenir sur son installation et assimiler des notions plus complexes une fois qu'il a atteint le niveau « utilisateur averti ».

Structure type d'un chapitre

La structure type d'un chapitre est la suivante :

- *Objectifs.* Cette partie énonce les objectifs principaux du chapitre. Les concepts sont abordés dans le paragraphe intitulé *Principe*.
- *Principe.* Ce paragraphe introduit les concepts qui seront abordés dans le chapitre. Il rappelle également dans quel contexte d'exploitation ou d'administration ils trouvent leur utilité. Si le contexte est trop général, ce paragraphe est omis car il ne sert à rien – c'est le cas dans le chapitre 1, qui

9782340-038004_001_828.indd 14 01/04/2020 16:34

présente l'historique du système UNIX et ses caractéristiques. Ce paragraphe peut se trouver en tête de chapitre ou débuter certaines sections importantes.

- *Corps du chapitre*. Cela concerne les développements théoriques et pratiques du thème abordé.
- *Résumé.* Nous proposons dans cette partie une synthèse du chapitre, en rappelant les concepts importants qui ont été évoqués.
- *Questions de révision.* Il s'agit d'une suite de questions qui concernent les points forts du chapitre. Cette partie doit permettre au lecteur de vérifier s'il a assimilé les concepts du chapitre. Les réponses sont en fin d'ouvrage.
- *Exercices.* Une liste d'exercices permet au lecteur de contrôler ses connaissances techniques et de mettre en pratique ses acquis. Les corrections sont en fin d'ouvrage.

Conventions typographiques

L'emploi de polices différentes permet de distinguer ce qui est saisi au clavier ou de faire ressortir un nouveau terme ou un nouveau concept :

- Les commandes ou leur résultat ont une police à espacement fixe, comme la commande `ls -l`. Les lignes de commandes sont présentées comme suit :

```
ls -l
```

- La première occurrence des termes nouveaux ou importants apparaît en italique, par exemple *inode*.

Le symbole ➥ en début de ligne indique que celle-ci et la ligne précédente ne sont qu'une seule et même ligne, comme par exemple :

```
dupont:x:1000:1101:Jacques DUPONT,Enseignant,20190621:
➥/home/crip/dupont:/bin/bash
```

Des encadrés de texte tramé apportent des informations complémentaires au fil des sections et des développements théoriques ou pratiques comme l'encadré « à noter » ci-dessous.

À NOTER

Cette rubrique attire l'attention du lecteur sur des points intéressants.

Les exemples

Les exemples fournis correspondent à des manipulations réalisées sur des serveurs et des postes de travail configurés puis paramétrés pour les besoins de l'ouvrage. L'utilisation des mêmes ordinateurs dans la partie « utilisation » et dans la partie « administration » maintient une cohérence entre les différents exemples, tout au long du livre. Dans la première partie, le lecteur trouvera des noms d'utilisateurs qui sont créés et modifiés dans la seconde partie. Cette unicité du contexte facilite la compréhension du niveau « administration », en présentant un environnement déjà vu dans la partie « utilisation ». Il est important de ne pas se contenter d'aborder des concepts en les agrémentant d'exemples sans liens entre eux, mais de proposer une mise en œuvre réelle de postes de travail et de serveurs, et de maintenir un fil conducteur tout au long de l'ouvrage.

Les exemples respectent des conventions typographiques. La police est à espacement fixe permet de distinguer la présence des espaces dans les commandes, ce qui peut s'avérer important. Les caractères gras sont utilisés pour distinguer le texte saisi par l'utilisateur ou les informations importantes. Les commandes sont généralement suivies du résultat de leur exécution, et précédées par l'invite système (`$` ou `#`).

À NOTER

Le caractère $ est la forme courante de l'invite dans les environnements bsh, ksh et bash du système UNIX. Le prompt # indique que les commandes sont saisies sous le compte *root* dans l'environnement Linux.

Questions de révision et exercices

Les questions de révision sont extrêmement utiles. Elles doivent permettre au lecteur de pointer les différents concepts abordés dans le chapitre et de vérifier qu'il les comprend bien. Cela fait partie des méthodes de travail de nombreux enseignants que de poser quelques questions ciblées à la fin d'un cours, en vue de vérifier si les étudiants ont assimilé les points essentiels, ou pour leur montrer qu'ils doivent en retravailler certains.

Nous proposons ensuite un ensemble d'exercices qui reprennent tous les points techniques traités dans le chapitre. Plusieurs réponses sont parfois possibles et seule une manipulation sur l'ordinateur peut confirmer ou infirmer la solution envisagée.

S'il n'est pas indispensable de réaliser tous les exercices, il est toutefois conseillé de s'atteler à quelques-uns, car rien ne remplace l'apprentissage par l'expérience. C'est ce qui explique le nombre important d'exercices proposés dans cet ouvrage.

Les réponses aux questions et les solutions des exercices sont fournies à la fin de l'ouvrage.

Contenu de l'ouvrage

L'apprentissage d'un système, quel qu'il soit, suit souvent la même logique. Après un bref rappel historique et un survol des caractéristiques du système, l'utilisateur apprend à gérer ses données, constituées de fichiers et de répertoires, puis à maîtriser son environnement de travail. Cela va de la notion de variables d'environnement à l'interface graphique, en passant par la protection de son espace personnel. Des thèmes plus complexes sont abordés, comme le réseau ou l'exécution de logiciels.

Aujourd'hui, l'administration d'un système n'est plus orientée vers l'espace personnel mais vers les services à offrir. Les thèmes récurrents sont donc l'installation, la gestion des utilisateurs, la gestion de l'espace disque et l'exploitation du serveur. Certains aspects revêtent une importance particulière, quand les serveurs sont plus importants en termes de nombre d'utilisateurs ou de services. C'est le cas des systèmes de fichiers et des services réseau.

Voici les concepts abordés dans les chapitres de la partie « utilisation » :

- Le **Chapitre 1** introduit le système UNIX. Il fait un rappel historique sur UNIX et Linux, et détaille les principales caractéristiques de ce système.
- Le **Chapitre 2** présente l'accès au système. Il évoque les différents modes de connexions et propose une initiation au réseau.
- Le **Chapitre 3** aborde la notion d'identification et de système multi-utilisateur. Il montre comment se connecter et se déconnecter, et indique quels fichiers du système identifient chaque utilisateur.
- Le **Chapitre 4** décrit les fichiers et les répertoires. Il présente la hiérarchie UNIX et détaille la syntaxe des commandes de base.
- Le **Chapitre 5** étudie la gestion des fichiers et des répertoires avec l'interface graphique.
- Le **Chapitre 6** porte sur la gestion avancée de l'espace disque. Il présente certaines commandes spécifiques et expose la façon d'accéder aux périphériques amovibles ou aux partitions Windows sous Linux. Enfin, il aborde la notion de systèmes de fichiers et de quotas disques, qui sont mis en œuvre dans la partie administration.
- Le **Chapitre 7** traite de la gestion des droits sur les fichiers et sur les répertoires. Il se termine par les droits d'endossement et les ACL.
- Le **Chapitre 8** porte sur l'environnement de travail. Il développe la notion de shells et de variables. Il aborde également les fichiers de démarrage.
- Le **Chapitre 9** introduit la notion de processus. Il explique l'organisation des processus en hiérarchie et montre comment les gérer. La fin du chapitre aborde les notions de traitement différé et de démons, utiles aux administrateurs.
- Le **Chapitre 10** traite du réseau. Il présente les commandes et les services. Il présente les commandes qui servent aux connexions à distance, aux transferts de fichiers, aux copies par le réseau ou aux connexions cryptées par SSH.

- Le **Chapitre 11** propose des commandes et des méthodes complémentaires : éditeur de texte `vi`, impression, sauvegarde, compression de fichiers. Il détaille aussi les processus de redirection des entrées/sorties, et l'aide en ligne.

Voici les concepts abordés dans les chapitres de la partie « administration » :

- Le **Chapitre 12** explique comment installer un système Linux. Il aborde le partitionnement et les étapes d'installation. Il indique comment installer des paquets logiciels ou des logiciels distribués au format source.
- Le **Chapitre 13** porte sur la configuration du système et sur l'accès au réseau. Il présente des outils d'administration comme `gnome-control-center`. Il développe les notions de classes d'adressage et de masques de sous-réseaux. Il détaille les fichiers de configuration d'un client DNS et du super-démon `inetd`. Il se termine par la configuration du service d'impression.
- Le **Chapitre 14** traite de l'exploitation d'un serveur. Il détaille la configuration du démarrage, le processus `init`, le démon `systemd`. Il présente l'arrêt et le démarrage des services, et donne un exemple de sauvegarde et de réinstallation d'un PC en amorçage multiple (*multiboot*) *via* le réseau. Enfin, il propose une solution de clonage de poste de travail avec les outils standard Linux.
- Le **Chapitre 15** porte sur la gestion des utilisateurs. Il explique comment créer et détruire des groupes et des utilisateurs, et expose la notion de *shadow password*. Il propose un script de création automatique de comptes utilisateurs.
- Le **Chapitre 16** aborde la gestion des systèmes de fichiers. Il en présente l'utilité et les différents types. Il détaille le montage, le démontage et la création de systèmes de fichiers, ainsi que la mise en œuvre d'espaces de swap. Il montre comment déployer les quotas disques et les volumes logiques.
- Le **Chapitre 17** présente les services réseau. Il indique comment installer des serveurs DNS, NFS et NIS sous Linux. Il traite également du serveur de messagerie. Il détaille l'accès à ces services à partir d'un poste client Linux.

Les suppléments

Le site **http://jeanmichel.lery.free.fr** propose le téléchargement des fichiers de cet ouvrage. Le mot de passe est : « **UnixLinuxUtilisationAdministration** ».

Remerciements

Je tiens à remercier Frédéric Jacquenod pour sa relecture avisée. Une attention toute particulière pour Sylvie, Guillaume et Julien.

PARTIE I – Utilisation

Chapitre 1 Présentation du système UNIX
Chapitre 2 L'accès au système
Chapitre 3 La première session
Chapitre 4 Les fichiers et les répertoires
Chapitre 5 Gestion des fichiers sous Gnome
Chapitre 6 Gestion de l'espace disque
Chapitre 7 Les droits
Chapitre 8 L'environnement de travail ou shell
Chapitre 9 Les processus
Chapitre 10 Le réseau
Chapitre 11 Compléments

9782340-038004_001_828.indd 19 01/04/2020 16:34

1

Présentation du système UNIX

PLAN

1. Bref historique d'UNIX
2. Les caractéristiques d'UNIX

OBJECTIFS

- Comprendre la genèse du système UNIX, et son incidence sur les différentes versions d'UNIX
- Assimiler les différences entre les versions d'UNIX
- Comprendre le lien entre UNIX et les logiciels libres
- Comprendre la notion de distribution Linux et de version du noyau
- Connaître les caractéristiques communes aux systèmes UNIX

UNIX est un système incontournable pour un administrateur. Il gère des ordinateurs départementaux, des stations de travail, mais aussi des micro-ordinateurs. Nous proposons dans ce chapitre un bref rappel historique, et présentons les principales caractéristiques qui font d'UNIX un système d'exploitation tout à fait adapté au monde des serveurs.

1. Bref historique d'UNIX

Cette partie présente la genèse du système UNIX, et son lien avec le monde des logiciels gratuits. Nous y apprenons comment Linux s'est développé sur la base d'un modèle de travail collaboratif unique dans son genre.

1.1. Histoire d'UNIX

UNIX découle du projet MULTICS (*Multiplexed Information and Computing Service*) initié en 1964 par le MIT (*Massachusetts Institute of Technology*), General

9782340-038004_001_828.indd 21 01/04/2020 16:34

Electric et les laboratoires BELL d'AT&T. Ce projet avait pour but de proposer un système fiable 24h sur 24h et 7j sur 7j, qui soit multi-utilisateur et avec lequel le travail en tâche de fond soit possible. En 1969, AT&T se retire du projet, mais deux ingénieurs décident de continuer et de développer, sur un PDP-7 (*Programmed Data Processor*) de DEC, un système plus simple que MULTICS. Ken Thompson et Denis Ritchie créent UNICS (*UNiplexed Information and Computing System*), et Brian Kernighan le renomme UNIX. Le développement du langage C par Denis Ritchie en 1972 a permis la réécriture du système UNIX en 1973, et son adaptation aux ordinateurs DEC PDP-11. Cela a été le véritable point de départ de son histoire : ce langage de haut niveau rendait UNIX portable, c'est-à-dire adaptable à tout ordinateur possédant un compilateur C.

À NOTER

Le langage C succède au langage B, développé par Ken Thompson pour l'ordinateur PDP-7 en 1970.

Un autre facteur favorable au développement du système UNIX a été la distribution gratuite des programmes sources par Thompson et Ritchie. La loi interdisait à la société AT&T (les anciens laboratoires Bell) de vendre des logiciels, du fait de son monopole sur les télécommunications. Cette distribution des sources a permis l'émergence d'une multitude de systèmes UNIX couramment appelés « UNIX-like » (AT&T ayant conservé le copyright sur UNIX). La distribution gratuite des codes sources, même si elle était initialement contrainte, a ouvert la voie aux logiciels libres que nous connaissons aujourd'hui.

En perdant son monopole, la société AT&T a pu commercialiser son UNIX sous l'appellation « UNIX System III » (puis System V), dont la dernière version de 1989 porte le nom de « System V Release 4 » (SVR4). Elle a également proposé la normalisation des appels systèmes, afin de permettre à tout logiciel écrit en langage C d'être portable sur n'importe quelle distribution UNIX par simple compilation. Cette norme SVID (*System V Interface Definition*) a été suivie par d'autres normalisations, telles OSF/1 de l'OSF (*Open Software Foundation*), ou POSIX (*Portable Operating System Interface for Unix*) de l'institut IEEE (*Institute of Electrical and Electronics Engineers*), ce qui confère un caractère « ouvert » au système UNIX. Par la suite, en 1991, la société AT&T a vendu ses droits sur UNIX à Novell, qui les a ensuite cédés à l'Open Group, émanation de l'OSF.

L'université de Berkeley a également contribué activement au développement d'UNIX avec sa distribution BSD (*Berkeley Software Distribution*), qui intégrait des fonctions comme la mémoire virtuelle, la pagination ou le protocole TCP/IP. En 1978, sous l'impulsion de Bill Joy, est sortie une nouvelle distribution d'UNIX, le 1BSD. Cette université est également à l'origine d'une importante distribution libre de droits, FreeBSD. Le système MacOS est basé initialement sur BSD.

Les grandes distributions UNIX comme AIX (*Advanced Interactive eXecutive*), HP-UX, Solaris et Linux sont compatibles System V et BSD. Elles implémentent

les évolutions techniques proposées par ces deux grandes « lignées » d'UNIX, et proposent des librairies de compatibilités spécifiques à l'une ou à l'autre, ce qui autorise l'installation de logiciels spécifiques.

À NOTER

Pour bien situer les événements, il est important de rappeler que le premier système multi-utilisateur, CTSS (Compatible Time Sharing System), développé par le MIT, est apparu en 1961. MS-DOS (MicroSoft-Disk Operating System) a vu le jour en 1981 ; le premier OS d'Apple, Mac OS 1.0 (sans parler de LisaOS), est sorti en 1984, et l'interface graphique Windows date de 1985.

1.2. UNIX et les logiciels libres

Dès le départ, les gènes d'UNIX intégraient la notion de logiciel libre. À l'origine, UNIX était distribué gratuitement par AT&T. Les centres de recherche et les universités comme Berkeley ont participé activement à son évolution, en développant de nouvelles fonctionnalités qui ont parfois été incluses dans l'UNIX d'AT&T. La notion de travail à la fois autonome et collaboratif, chère aux logiciels libres, était née.

En réaction à la commercialisation d'UNIX par AT&T en 1983, plusieurs mouvements de développeurs de logiciels libres de droits ont vu le jour. Ainsi, en 1984, Richard Stallman, du MIT (*Massachusetts Institute of Technology*), a initié un mouvement qui a abouti à la création de la FSF (*Free Software Foundation*). Cette fondation a développé les projets GNU, à l'origine de logiciels libres comme GNU Emacs (éditeur de texte), GNU C (compilateur C), GNU Bash (interpréteur de commandes), etc. Les logiciels de cette famille utilisent comme symbole une tête de gnou, et le terme GNU est un acronyme récursif (défini par lui-même) qui signifie *Gnu's Not Unix*. Les logiciels GNU sont distribués sous la licence GPL (*Gnu Public Licence*), laquelle stipule que toute personne peut utiliser gratuitement le logiciel, et même le commercialiser. Mais toute modification doit elle-même être distribuée gratuitement à la communauté. Ce mode de distribution est un réel facteur de développement des logiciels libres. Linux a adopté la licence GPL.

Certains mouvements se sont concentrés sur une partie du système UNIX. C'est le cas du *Consortium X* (du MIT), qui a fait évoluer le protocole graphique X-Window (ne pas confondre avec Windows !) sur la demande de la communauté UNIX. Par la suite, le groupe Xfree86 a porté ce système graphique sur les plates-formes à base de processeur INTEL (*INTegrated ELectronics*). Cette version pour micro-ordinateur, nommée XFree86, est disponible gratuitement. Elle est intégrée dans les distributions Linux.

1.3. Histoire de Linux

Linux est un système UNIX adapté aux micro-ordinateurs à base de processeurs INTEL. Il a été créé par Linus Torvalds, étudiant à l'université d'Helsinki, en août

1991. (Le terme Linux doit son origine à Ari Lemmke, administrateur du tout premier site de distribution Linux, ftp.funet.fi.) Cet étudiant finlandais de 21 ans voulait développer un système utilisant au mieux les fonctionnalités multitâches du processeur Intel 80386. Il s'est inspiré du système Minix, développé par Andrew S. Tanenbaum et ses étudiants en 1986-1987, qu'il avait installé sur son micro-ordinateur mais qu'il trouvait trop limité. Linus Torvalds disait de son système Linux (Linus'UNIX) : « *better Minix than Minix* ». C'est tout naturellement qu'il en a fait l'annonce sur le forum USENET réservé à Minix, « comp.os.minix ». Il a ensuite lancé un appel à contribution et donné libre accès au code source.

La première version (0.0.1) d'août 1991 était rudimentaire et nécessitait l'installation de Minix pour être compilée. Linux devient autonome le 5 octobre 1991, avec la sortie de la version 0.0.2, première version « officielle ». Elle permet d'utiliser des logiciels GNU, comme le compilateur *gcc*, la commande *make*, le filtre *sed*, le shell *bash*. Linus Torvalds décide, par la suite, de distribuer Linux sous licence GPL. Cette nouvelle méthode de développement à laquelle tout le monde peut participer est la grande force du système : le logiciel n'est plus produit par une équipe isolée mais par une communauté de développeurs, répartis dans le monde entier, et qui s'échangent des idées et des codes sources *via* le réseau Internet. Aucun organisme particulier ne contrôle le développement, mais Linus Torvalds est responsable du développement du noyau (« *kernel* »), et il décide de la distribution des codes sources. Le développement de chacun des composants (gestion de la mémoire, système de fichiers, etc.) est géré par un chargé de projet qui collecte les contributions des autres développeurs et les soumet à Linus Torvalds pour qu'il les intègre au noyau.

De 1991 à 1994, Linux est porté sur d'autres plates-formes comme des ordinateurs à base de processeur Alpha. Durant cette période, le système de fichiers de Linux, hérité de Minix, est remplacé par ext2fs (*Second Extended Filesystem*), ce qui le fait évoluer vers un système fiable et professionnel. La version 1.0 est diffusée *via* Internet le 14 mars 1994. En 1995, l'explosion du réseau Internet donne une forte impulsion au système Linux. D'une part, la communauté des développeurs s'élargit de plus en plus, mais, surtout, un nombre important de fournisseurs d'accès à Internet le choisissent pour sa fiabilité, sa robustesse et sa gratuité. Il devient incontournable dans l'administration des serveurs Web, associé au logiciel *Apache*. Cette utilisation de Linux comme système professionnel est renforcée par l'apparition de sociétés commerciales telles que Red Hat, qui le distribuent gratuitement et proposent des interfaces d'installation simplifiées ainsi que des services d'aide aux entreprises. En 1995, la société Red Hat Software sort la version 2.0 de son Linux, dans laquelle est intégré l'outil de gestion simplifié des paquetages logiciels RPM (*Red Hat Package Management*).

Dès 1996, Linux apparaît sur tous les fronts. Du côté des entreprises, des sociétés comme Netscape et Corel commencent à distribuer des versions Linux de leurs produits. Quant au grand public, il n'est pas oublié, avec l'évolution de l'interface

9782340-038004_001_828.indd 24 01/04/2020 16:34

graphique et le début du projet de bureau KDE puis du projet concurrent GNOME. Enfin, les années 2000 consacrent Linux auprès des grandes sociétés informatiques. Sun propose une suite bureautique gratuite, StarOffice 5.2, puis fournit les codes sources pour OpenOffice.org. Dès 2002, IBM et HP présentent des solutions professionnelles de leurs serveurs, basées sur Linux.

1.4. Les versions du noyau de Linux

La numérotation des versions de Linux indique son niveau de développement, son état de stabilité et l'évolution des correctifs.

La convention de numérotation a évolué depuis le noyau 2.6, elle possède désormais la forme : w.x.y.z. La valeur de z caractérise une correction minime d'erreur de conception ou « bug ». À chaque correction, z est incrémenté. La valeur de y informe sur la version de mise à jour du noyau w.x. Une valeur *paire* de x désigne une version stable. Une valeur *impaire* s'applique à une version de développement, également appelée version en *bêta test*. Ainsi, le numéro 2.6.11 représente la 11^{e} version de mise à jour du noyau stable 2.6. Après la version 2.6 cette régle avec les valeurs paires et impaires a été progressivement abandonnée. En août 2019, la version stable du noyau est 5.2.9.

1.5. Les distributions Linux

Le système Linux est actuellement fourni sous forme de *distributions*. Celles-ci comprennent un noyau à une certaine version, des pilotes de périphériques, des outils d'installation et d'administration, ainsi que différents logiciels. Nous décrivons ici les principales distributions Linux, avec leurs caractéristiques :

- **Slackware.** C'est une distribution non commerciale ancienne, proche de la « philosophie originale d'UNIX ». Elle est utilisée par les développeurs « irréductibles ». Site de référence : **www.slackware.com**.
- **Red Hat Enterprise Linux.** C'est une distribution commerciale. L'installation des logiciels y est simplifiée grâce au gestionnaire de paquetages RPM (*Red Hat Package Manager*). Ces paquetages fournissent les logiciels déjà compilés, la documentation associée, ainsi qu'un système de dépendance permettant d'installer automatiquement les autres logiciels nécessaires. Site de référence : **www.redhat.com**.
- **Debian.** Cette distribution non commerciale a su garder « l'esprit » initial des logiciels GNU. L'intervalle de temps entre deux versions stables est parfois plus important qu'avec d'autres distributions. Cette distribution utilise le gestionnaire de paquetages APT (*Advanced Packaging Tool*). Site de référence : **www.debian.org**.
- **SuSE.** Cette distribution d'origine allemande est basée sur la distribution Slackware. Elle convient à la fois aux utilisateurs néophytes et aux administrateurs qui ne veulent pas trop intervenir au cœur du système. Elle utilise

les paquetages de la Red Hat. Elle est connue pour sa distribution commerciale SuSe Linux Enterprise, mais propose aussi une version libre de droits OpenSUSE. Site de référence : **fr.opensuse.org**.

- **Fedora.** Cette distribution est une version gratuite soutenue par Red Hat. Site de référence : **www.fedora-fr.org.**
- **CentOS.** C'est une version gratuite dérivée de la Red Hat. Elle est entière compatible avec la version commerciale Red Hat Enterprise Linux. Site de référence : **www.centos.org.**
- **Ubuntu.** Cette distribution basée sur la distribution Debian reprend les différents outils de celle-ci et propose un installateur Graphique. Il existe une version commerciale et une version gratuite. La version LTS (*Long Term Support*) est la version stable. Site de référence : **www.ubuntu-fr.org**.
- **Knoppix.** Cette distribution « Live CD » ne nécessite aucune installation sur le disque dur. Il suffit de démarrer sur le CD-ROM ou une image disque pour se retrouver sous Linux. Elle permet à tout utilisateur de tester Linux sans qu'il ait à reconfigurer son disque dur. Elle est basée sur une distribution Debian compressé. Site de référence : **www.knoppix.org**.
- **Linux Mint.** Cette distribution gratuite est basée sur Ubuntu. Elle est conçue pour les ordinateurs individuels. Site de référence : **linuxmint.com**.
- **Mageia.** Cette distribution gratuite française est basée sur Mandriva Linux qui a disparue. Elle est appréciée des débutants car elle propose des outils graphiques de configuration ainsi que la plupart des bureaux comme KDE, GNOME, XFCE. Site de référence : **www.mageia.org**.

Le choix d'une distribution est toujours délicat. Indiquer que telle distribution est meilleure que telle autre relève souvent d'une « guerre de chapelle ». Les utilisateurs devenus administrateurs préconiseront souvent la distribution qu'ils maîtrisent et sur laquelle ils ont appris Linux. Néanmoins, certaines distributions conviennent davantage aux utilisateurs, comme les distributions Ubuntu et Linux Mint, et d'autres sont plutôt recommandées pour les administrateurs, comme Red Hat Enterprise Linux, Debian, SuSE et FreeBSD (qui n'est pas une distribution Linux). Cela peut laisser penser que lorsqu'on est administrateur, l'interface graphique ne doit pas être un critère de sélection. Il est vrai que ce type d'interface est à privilégier pour les utilisateurs, lesquels choisiront une distribution comme Ubuntu : l'administration n'impose pas la saisie systématique des commandes et n'interdit pas les interfaces graphiques. Or l'interface graphique est bien un critère à prendre en compte lorsqu'on choisit une distribution pour administrer un serveur. Ainsi, le système AIX (UNIX IBM) dédié aux serveurs propose, depuis ses origines, une interface de ce type (SMIT, pour *System Management Interface Tool*) très appréciée des administrateurs. Quant aux distributions Red Hat Enterprise Linux et Ubuntu, elles sont tout autant adaptées aux utilisateurs qu'aux administrateurs. Si la grande majorité des distributions propose aux utilisateurs des

interfaces graphiques équivalentes avec les bureaux KDE et GNOME, elles sont inégales sur le plan de l'interface graphique dédiée à l'administration.

2. Les caractéristiques d'UNIX

Il existe plusieurs versions commerciales du système UNIX, et différentes distributions du système Linux. Si toutes ces versions portent des noms distincts, elles partagent les mêmes caractéristiques et sont considérées comme des variantes d'un même système, UNIX. Elles ont en commun les caractéristiques des deux grandes lignées que sont UNIX System V (société AT&T) et BSD (Université de Berkeley).

2.1. Un système multiplateforme

UNIX est le seul système d'exploitation au monde que l'on peut trouver sur tous les types d'ordinateurs, du serveur au poste de travail, et ce, quel que soit le processeur (Intel, ARM, etc.). Développé à l'origine pour les serveurs utilisés dans le monde de la recherche et des télécommunications, il a vite été adopté par les grandes sociétés commerciales comme IBM, HP, SUN, pour équiper leurs serveurs généralistes. UNIX gère aussi bien les supercalculateurs que les stations de travail graphiques ou scientifiques.

Ce système est également présent dans le monde de la micro-informatique, du fait des nombreuses distributions Linux adaptées aux compatibles PC. L'évolution des interfaces graphiques et l'apparition de suites bureautiques libres de droits en font un système d'exploitation tout à fait adapté aux postes de travail dédiés à la bureautique. Le développement d'Internet a par ailleurs favorisé le déploiement de Linux sur les micro-ordinateurs et, de là, l'émergence de petits serveurs Web fiables et peu chers. Enfin, le système d'exploitation de la société Apple, Mac OS X – devenu MacOS –, qui est apparu en 2002, est entièrement basé sur un noyau UNIX issu de FreeBSD et de NetBSD. Aujourd'hui, UNIX est un système d'exploitation incontournable pour un administrateur système.

2.2. Un système multitâche et multi-utilisateur

UNIX a été conçu pour autoriser les connexions multiples sur un seul ordinateur. Comme il gère plusieurs sessions d'utilisateurs simultanées, on dit qu'il est *multi-utilisateur*. Chaque utilisateur se connecte au système par l'intermédiaire d'un terminal, ou d'un micro-ordinateur « transformé » en terminal, en faisant appel à un logiciel nommé *émulateur de terminal*. UNIX est en outre un système *multitâche* : une fois connecté, chaque utilisateur peut exécuter plusieurs logiciels simultanément.

2.3. Une hiérarchie arborescente gérée en « système de fichiers »

La gestion de l'espace disque et du stockage des données repose sur des *fichiers*, qui sont rangés dans des *répertoires*. Cette organisation en cascade – les répertoires peuvent contenir d'autres répertoires et des fichiers –, dont le point de départ est un répertoire « racine », porte le nom de *hiérarchie arborescente*. Dans cette représentation, un fichier est une feuille de l'arbre, et un répertoire (également appelé « nœud ») correspond à une branche.

Windows possède une structure identique. La différence principale tient à la gestion des disques durs : dans l'environnement UNIX, il n'y a pas de nom pour les unités disques, alors que c'est le cas sous Windows (C:, D:, etc.). La hiérarchie UNIX est déployée sur l'ensemble des disques, comme un « filet » recouvrant l'ensemble des unités de stockage. Les branches de l'arborescence sont réparties sur les différents disques ou partitions de disques. Cet arbre constitué de plusieurs branches autonomes mais liées entre elles, représente une seule entité logique pour l'utilisateur.

L'association entre une branche et un disque se nomme un *système de fichiers*, ou *FS*, pour *File System*. Un système de fichiers peut être monté ou démonté (on monte ou démonte la branche de l'arbre), ce qui active ou désactive l'accès à une partie de l'arborescence. Il est aussi possible de remonter un *file system* sur une autre partie de l'arbre, ce qui déplace toute une partie de la hiérarchie, moyennant quelques manipulations. Ce principe de la hiérarchie arborescente constituée de l'association de plusieurs branches autonomes et localisées sur des disques différents a été étendu au réseau. Un *file system* peut être présent sur un disque local de l'ordinateur ou sur un disque distant qui se trouve sur un autre ordinateur. Ce système de fichiers qui utilise le réseau est appelé « système de fichiers NFS » (*Network Filesystem*) .

2.4. Différents environnements de commandes : les shells

Un *shell* est un logiciel qui fait l'interface entre l'utilisateur et le cœur du système d'exploitation (le noyau ou *kernel*). Par exemple, il lit la commande saisie par l'utilisateur et transmet la demande d'exécution au noyau.

Contrairement à Windows, qui ne possède qu'une interface graphique, UNIX propose différents shells. Ces shells sont avant tout des interfaces en ligne de commandes (mode texte), mais il existe également des interfaces graphiques comme X-Window et, par-dessus ces interfaces, des bureaux graphiques tels que KDE (*K Desktop Environnement*) et GNOME (*GNU Network Object Model Environment*). Le terme « shell sous UNIX » s'applique principalement aux interfaces en ligne de commandes. La traduction de shell en français est « interpréteur de commandes ». L'utilisateur peut choisir entre le Bourne Shell (bsh), le Korn Shell (ksh), le C-Shell (csh), le Bourne Again Shell ou shell Linux (bash), et d'autres shells moins courants.

9782340-038004_001_828.indd 28 01/04/2020 16:34

Ces shells sont de véritables langages de programmation, qui permettent de développer en utilisant des commandes. Pour un administrateur, il est courant de développer des programmes en langage shell, appelés *scripts shell*. Cela lui permet d'enchaîner et d'automatiser des commandes pour des traitements systématiques. Le choix d'une interface particulière (graphique ou en ligne de commandes) change totalement la présentation et la perception du système UNIX. L'utilisateur est complètement libre de paramétrer son environnement, et ce, indépendamment du paramétrage des autres utilisateurs du même poste de travail ou du même serveur.

2.5. Gestion hiérarchisée des processus

Les *processus* correspondent aux tâches ou logiciels qui sont en cours d'exécution. Pour gérer ces applications qui se déroulent simultanément, UNIX définit une *hiérarchie de processus*. Au démarrage de l'ordinateur, un processus initial (init) est activé. À son tour, il lance l'activation des différents logiciels de gestion du système (gestion des disques, gestion du réseau, gestion des connexions, etc.), qui eux-mêmes lancent d'autres tâches propres à chaque utilisateur.

Dans cette hiérarchie, un processus qui en engendre un second est appelé *processus père*. Le processus engendré est appelé *processus fils*. Cette hiérarchie de processus est donc une hiérarchie de processus pères et de processus fils. Par exemple, lorsqu'un utilisateur est connecté, il exécute son shell (ksh, etc.). Quand il saisit la commande `vi`, il crée un nouveau processus à partir de son shell. Le processus « ksh » est le père du processus « vi » et, à l'inverse, « vi » est le fils de « ksh ». L'utilisateur peut arrêter temporairement, redémarrer ou terminer un processus dont il est le propriétaire (qu'il a initié). L'administrateur peut faire de même sur l'ensemble des processus. Un processus père peut être à l'origine de nombreux processus fils et, à leur tour, ces processus fils peuvent avoir de nombreux processus fils… Si vous supprimez un processus, vous supprimez toute sa « descendance » et arrêtez par conséquent les programmes associés à ces processus. Exemple : si vous détruisez le processus init (père de tous les processus), vous arrêtez votre ordinateur.

2.6. Un système ouvert

Un des points forts du système UNIX est d'avoir été normalisé à plusieurs reprises. Ainsi, l'OSF a proposé OSF/1. Ensuite, l'Open Group, qui a la charge de la normalisation dans le monde des systèmes ouverts, a proposé UNIX 95, puis UNIX 98. Quant au comité IEEE, il a défini la norme POSIX. Toutes ces normalisations offrent des bases homogènes aux constructeurs qui, en les respectant, développent des systèmes UNIX « semblables » dans leur utilisation ainsi que des logiciels plus facilement portables puisque les bases de constructions sont communes.

2.7. Différents UNIX : AIX (IBM), HP-UX (HP), Solaris (Oracle), Linux

Il n'existe pas un UNIX mais des « UNIX-like ». UNIX étant une marque déposée de l'Open Group, pour obtenir l'autorisation d'utiliser cette marque il est nécessaire de respecter la conformité à la *Single UNIX Spécification*. Le suffixe « ix » ou « ux » est souvent utilisé (sauf pour Solaris) pour rappeler que le système en question est un « UNIX ». Ces différents systèmes respectent la plupart des normes et s'appuient sur les systèmes System V et BSD. Si les différences sont quasi inexistantes pour un utilisateur final, elles sont bien réelles pour un administrateur : elles portent sur les commandes d'administration, tandis que les concepts sont identiques.

2.8. Un lien fort avec le langage C

À l'origine écrit en assembleur, UNIX a été récrit en C dès l'apparition de ce langage, en 1972. L'ajout de nouvelles fonctionnalités écrites en C n'a fait que renforcer le lien très fort qui existe entre ce langage et le système. Être administrateur système sous UNIX suppose un jour ou l'autre que l'on doive adapter ou « porter » un logiciel écrit en langage C. L'apprentissage du langage C et du système UNIX sont souvent liés, et la programmation C s'apprend généralement sur un système UNIX.

Résumé

UNIX est un système d'exploitation assez ancien qui a été développé pour des besoins internes par un groupe de chercheurs. Dès l'origine, il a été conçu pour gérer des serveurs, ainsi que des stations de travail sur lesquelles plusieurs utilisateurs devaient se connecter simultanément. Il a très vite été porté sur différents ordinateurs grâce au langage C, qui a servi de support à son développement. L'intérêt des constructeurs d'ordinateurs pour ce système n'a cessé de croître, à tel point qu'actuellement, les grands acteurs commerciaux proposent tous, avec leurs serveurs, un système UNIX basé sur l'une ou l'autre des deux grandes lignées d'UNIX, BSD de l'université de Berkeley, et System V de AT&T.

Ce système fascinant a ensuite été adapté au monde des micro-ordinateurs pour gérer au mieux leur mode multitâche. Le système Linux est né de la volonté de Linus Torvalds de proposer un système encore meilleur que le système Minix. Très vite, les utilisateurs et les développeurs de tout genre se sont appropriés ce système ; dès le début, les codes sources ont été distribués gratuitement. Le mode de travail collaboratif, représenté par la licence GPL actuelle, était né. Linux continue à se développer en utilisant les logiciels gratuits, comme les applications GNU.

Plusieurs caractéristiques font que ce système est devenu incontournable pour les administrateurs. La raison principale tient sans doute dans le fait qu'UNIX est le seul système que l'on puisse trouver sur tous les types d'ordinateurs. Les autres caractéristiques comme la gestion de l'espace disque en hiérarchie arborescente, la gestion du multitâche et des utilisateurs multiples, l'ouverture vers les normes, la possibilité pour chaque utilisateur de choisir son interface, ainsi que l'intégration du langage C, en font un système très apprécié des développeurs, mais aussi des utilisateurs, qui découvrent une autre façon de gérer leur micro-ordinateur, et un univers où tous les logiciels sont gratuits.

Le seul véritable problème est parfois de se repérer parmi les différentes distributions. Mais rien n'empêche d'opter pour une distribution particulière, en considérant que les autres offrent une liberté de choix supplémentaire.

Questions de révisions

1. Qui a créé UNIX et en quelle année ?
2. Quelle est l'incidence de l'invention du langage C pour UNIX ?
3. Comment se nomme le système UNIX gratuit proposé par l'université de Berkeley ?
4. Quel rôle important a joué la FSF (*Free Software Foundation*) dans le développement des logiciels libres et, indirectement, de Linux ?
5. Qui est le créateur de Linux ?
6. Quel est le nom du système UNIX qu'il utilisait alors et qui lui servait de référence pour produire un système encore meilleur ? Qui avait créé ce système ?
7. Quel est le mode de fonctionnement pour le développement du noyau Linux ?
8. Quelle est la différence entre la notion de distribution Linux et la notion de version commerciale d'UNIX ?
9. Qu'est-ce qu'un système multitâche ?
10. Qu'est-ce qu'un système multi-utilisateur ?
11. Qu'est-ce qu'un shell UNIX ?
12. Quel système de gestion est utilisé pour les fichiers et répertoires ?
13. Qu'est-ce qu'un *file sytem* ?
14. Qu'est-ce qu'un processus ?

Exercices

Les exercices qui suivent portent uniquement sur des questions de cours. Les exercices pratiques seront abordés à partir du chapitre suivant.

1. Quelles sont les deux grandes lignées du système UNIX ?

2. Citez trois noms de systèmes UNIX commerciaux (de grandes sociétés informatiques) et deux noms de distributions Linux.
3. Comment se nomme le logiciel gratuit qui adapte le protocole X-Window aux cartes graphiques des PC ?
4. Citez quatre shells UNIX.
5. Comment appelle-t-on un processus qui en exécute un autre ?

2

L'accès au système

PLAN

1. La connexion directe du poste de travail
2. L'accès par le réseau

OBJECTIFS

- Intégrer quelques notions de base sur le réseau
- Assimiler la notion d'adressage IP et de DNS
- Aborder la notion de terminal

Le système UNIX est multi-utilisateur, c'est-à-dire que plusieurs personnes peuvent travailler simultanément sur le même ordinateur. Si cela est moins évident avec Linux, qui est une adaptation d'UNIX au monde des PC, donc aux postes « mono-utilisateurs » sur lesquels une seule personne travaille à la fois, il est fréquent de trouver des sessions multiples sur les serveurs UNIX. Chaque utilisateur dispose de son propre poste de travail connecté au serveur. Cette connexion peut être directe (via un lien série) ou utiliser le réseau. Nous présentons dans ce chapitre les différentes méthodes d'accès, ainsi que les catégories de postes de travail.

1. La connexion directe du poste de travail

Cette partie aborde la notion de terminal et d'émulation de terminal.

1.1. Principe

À l'origine, l'accès aux serveurs UNIX utilisait des postes de travail appelés *terminaux*, qui étaient connectés directement aux interfaces séries (ports de communication) de l'unité centrale. Le nombre de ports séries disponibles à l'arrière du serveur limitait le nombre de terminaux que l'on pouvait installer et,

9782340-038004_001_828.indd 33 01/04/2020 16:34

par conséquent, le nombre de sessions simultanées. Aucun réseau d'accès n'était utilisé. La figure 2.1 présente ce mode de connexion.

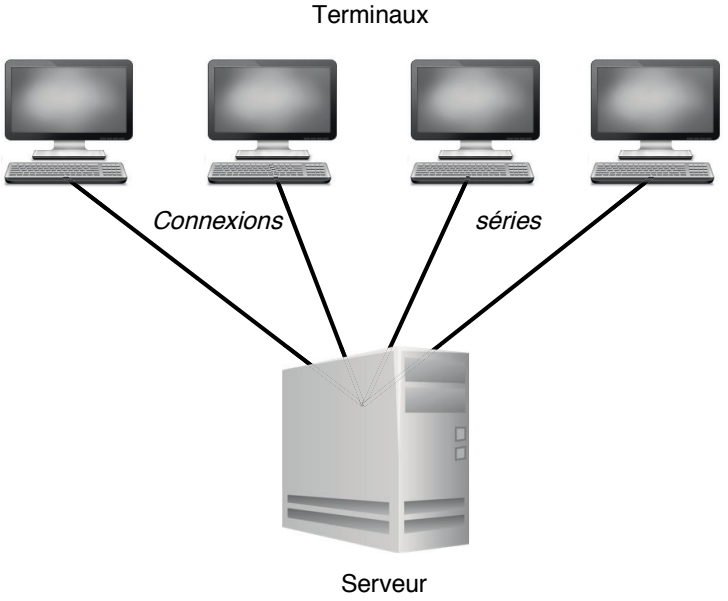


Figure 2.1
Terminaux connectés en direct.

Le principal défaut de cette méthode d'accès est son manque de souplesse (un terminal ne donne accès qu'au serveur sur lequel il est branché). Cependant, la confidentialité de l'information qui circule entre le terminal et le serveur sans être propagée vers les autres postes de travail, est un atout majeur. Ce type de connexion est toujours utilisé aujourd'hui (en nombre très limité), et il assure à l'administrateur un bon niveau de sécurité. Il présente un second avantage : l'accès au serveur est garanti, même en cas de dysfonctionnement du réseau. Cet atout devient déterminant quand le serveur lui-même en est la cause, par exemple, suite à une panne ou à une mauvaise configuration de ses éléments réseaux (cartes, protocoles). Cette connexion directe est aussi très importante pendant l'installation d'un serveur, le réseau n'étant alors pas encore configuré.

1.2. La notion de terminal

Les terminaux employés dans le cas d'une connexion sur un port série sont uniquement des périphériques d'entrée/sortie (E/S). Ils sont constitués d'un écran et d'un clavier, ce qui les différencie des micro-ordinateurs, qui possèdent en plus une unité centrale. Leur rôle est passif : tout ce qui provient du serveur s'affiche sur l'écran, tout ce qui est saisi au clavier est transmis directement au serveur (voir figure 2.2).

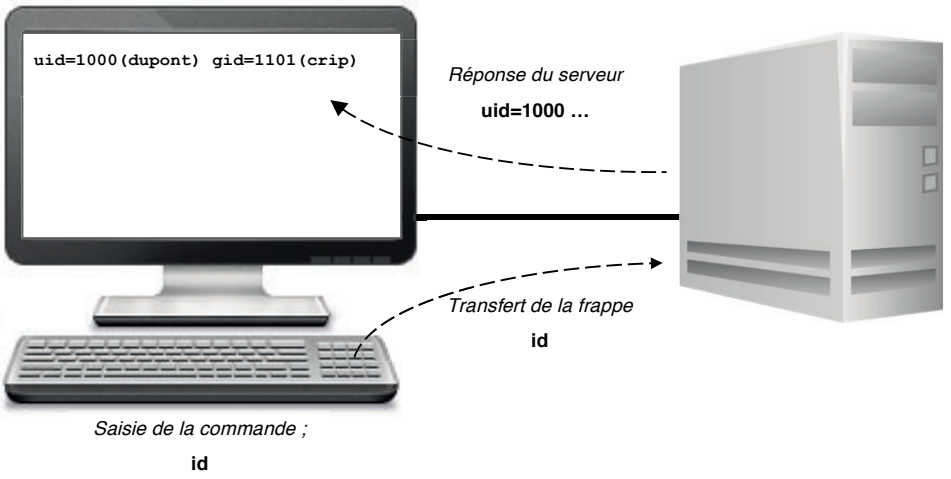


Figure 2.2
Fonctionnement d'un terminal.

La principale différence entre un terminal et le couple écran-clavier d'un micro-ordinateur est que le terminal est autonome et indépendant du système d'exploitation. Il peut être connecté à n'importe quel serveur, même si celui-ci n'est pas géré par UNIX. Il dispose d'un langage de contrôle appelé *suite d'échappement* qui permet au serveur de le piloter grâce à des ordres (il peut ainsi commander l'effacement d'écran). Par ailleurs, le terminal est configurable (vitesse de communication, port de connexion, touches d'échappement, langue).

Dans le cas du micro-ordinateur, l'écran et le clavier sont connectés à l'unité centrale. L'écran est directement contrôlé par la carte graphique qui s'y trouve, et non par un langage de contrôle. Il ne peut se connecter que sur un autre micro-ordinateur possédant une carte graphique compatible (ce qui est devenu courant aujourd'hui). Si l'affichage sur un terminal pose problème, vous pouvez le redémarrer pour réinitialiser la connexion. Par contre, dans le cas d'un écran de micro-ordinateur, c'est tout l'ordinateur qu'il faut redémarrer, car l'affichage dépend directement de la carte graphique qui se trouve dans l'unité centrale.

1.3. L'émulation de terminal

Il est tout à fait possible de brancher un micro-ordinateur directement sur le port série du serveur UNIX. Il faut pour cela un logiciel qui le transforme en terminal. Ce type de logiciel porte le nom d'*émulateur de terminal*. Il rend « transparente » l'unité centrale du micro-ordinateur, qui devient un simple périphérique d'entrée/sortie, et lui « apprend » le langage de contrôle du terminal émulé. Chaque terminal possède son propre langage en suite d'échappement, dont les fonctionnalités dépendent de ses caractéristiques texte ou graphiques. Les terminaux supportés par les émulateurs se répartissent en trois grandes familles :

- **Les terminaux ASCII.** Dans cette famille, le constructeur DEC faisait référence, avec ses modèles vt100, vt220, vt340. Le terminal DEC vt100 a même fourni la base du langage de terminal normalisé par l'organisme américain ANSI (*American National Standards Institute*).
- **Les terminaux graphiques.** Dans cette famille, le constructeur Tektronix était la référence, avec ses terminaux monochromes TEK4010, TEK4014 ou son terminal couleur TEK4207.
- **Les terminaux X.** Cette catégorie est abordée ultérieurement. Il s'agit d'un type de terminal graphique et réseau respectant le protocole graphique X-Window (à ne pas confondre avec Windows !).

Voici comment le serveur utilise le langage du terminal pour en contrôler l'affichage (voir figure 2.3). En saisissant la commande UNIX `clear`, on demande au serveur UNIX d'effectuer un effacement d'écran. Le serveur envoie cette instruction au terminal dans son langage, sous la forme : `Esc[2J` (caractère Escape, crochet gauche, 2 et J majuscule). Le terminal analyse le texte à afficher, détecte et interprète la séquence de contrôle, puis procède à l'effacement.

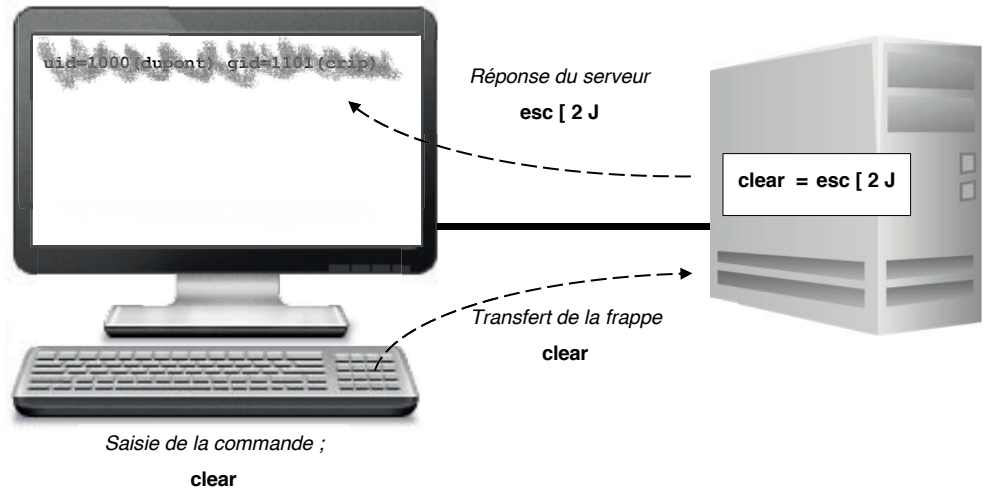


Figure 2.3
Langage de commande du terminal.

1.4. Le terminal sous Linux : la console

Un PC administré sous Linux se comporte un peu différemment. Si la notion de terminal est bien présente, du fait de la possibilité de contrôler l'écran par des suites d'échappement, l'écran et le clavier du micro-ordinateur branchés sur l'unité centrale ne constituent pas vraiment un terminal (l'écran est piloté par la carte graphique de l'unité centrale, il n'est donc pas autonome). C'est également le cas des écrans graphiques que l'on trouve sur les serveurs UNIX utilisés comme stations de travail graphiques. On ne parle plus alors de terminal mais de *console*.

La console correspond à un type particulier de terminal émulé. Ainsi, sous AIX (UNIX IBM), le nom du terminal est `dtterm` et il est en partie compatible avec le terminal DEC vt100. La sélection d'un mauvais type de terminal peut engendrer des problèmes d'affichage ou la non-reconnaissance des touches de déplacement (voir chapitre 3).

2. L'accès par le réseau

Cette partie présente les méthodes actuelles d'accès par le réseau et aborde la notion de client/serveur.

2.1. Principe

Depuis l'apparition du réseau, la connexion directe a été progressivement remplacée par d'autres techniques qui utilisent ce support (voir figure 2.4). Dans un premier temps, la connexion série a été déplacée de l'arrière du serveur vers la salle de travail des utilisateurs, grâce à un boîtier qui possédait d'un côté une prise permettant de le brancher sur le réseau, et de l'autre un ensemble de ports série (connectés aux postes de travail). Le câble réseau passait dans chaque salle où se trouvait ce boîtier sur lequel tous les terminaux étaient connectés. Ce type de matériel se nommait un *concentrateur de terminaux* et préfigurait les *concentrateurs réseaux* ou *hubs* d'aujourd'hui.

Ensuite sont apparus des terminaux ayant leur propre carte Ethernet. Ceux-ci sont directement connectés au réseau et peuvent atteindre n'importe quel serveur qui se trouve sur le réseau mondial. Cette méthode autorise l'accès à une multitude de serveurs géographiquement distants à partir d'un seul poste de travail, et l'affichage de chaque session dans une fenêtre graphique dédiée. C'est le cas des *terminaux X*.

Enfin, on peut se servir des micro-ordinateurs disposant d'une carte Ethernet comme de terminaux, si l'on utilise un émulateur de terminal communiquant avec le serveur *via* le réseau. Le protocole Telnet (*TELecommunications NETwork*) s'appuie sur le concept de *terminal réseau virtuel* ou NVT (*Network Virtual Terminal*). Le protocole SSH (*Secure Shell*) propose une connexion sécurisée. Les commandes qui implémentent ces protocoles sous UNIX sont `telnet` et `ssh`. Sous Windows le logiciel libre, `putty.exe`, offre la possibilité d'utiliser des connexions cryptées *via* ssh. Le protocole SSH est généralement préféré à Telnet. Il présente en effet l'avantage de crypter la communication *via* des clés de chiffrement et propose une meilleure sécurité.

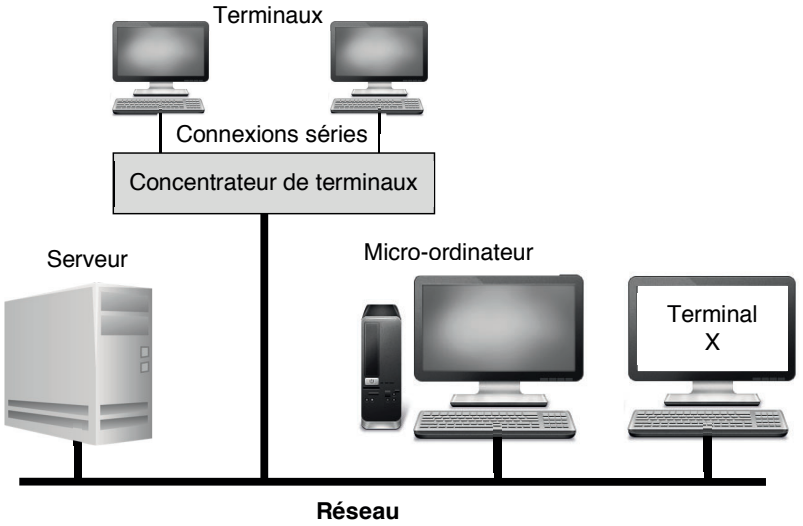


Figure 2.4
Les techniques d'accès par le réseau.

De nos jours, la méthode d'accès « réseau » la plus courante est l'émulation de terminal sur un micro-ordinateur, et plus particulièrement le recours à un client Telnet (connexion non cryptée) ou ssh (connexion cryptée). Cette solution offre plus de possibilités avec un poste de travail « intelligent », qui autorise des fonctions comme la copie d'écran ou le copier-coller entre la fenêtre du terminal et celle d'un traitement de texte s'exécutant en local. Cependant, elle nécessite des connaissances de base sur les réseaux.

Lorsqu'on dispose d'un simple terminal connecté en direct, il suffit de le mettre sous tension pour établir la connexion. Dans le cas du réseau, il faut assimiler certaines notions comme l'adressage IP (*Internet Protocol*) ou le rôle d'un serveur DNS (*Domain Name System*). Généralement, la commande `telnet`, appliquée au nom (complet) du serveur à atteindre, suffit à établir la connexion. Mais parfois cela ne fonctionne pas, et quelques notions sur les réseaux suffisent à résoudre le problème. Nous étudions ci-après les principes de base de la communication entre les serveurs et les postes de travail sur un réseau. Nous abordons notamment les notions d'adresses IP et de serveurs DNS.

2.2. Adresses IP et DNS

Tout ordinateur ou terminal connecté au réseau possède une adresse IP qui lui permet d'établir une communication avec un autre ordinateur. Dans l'univers informatique, elle est l'équivalent de notre adresse postale. Les adresses postales nous permettent aussi bien d'envoyer des courriers à des correspondants que de nous rendre chez des amis ou de recevoir un catalogue de vacances. De la même manière, les ordinateurs utilisent l'adressage IP pour faire parvenir des courriers

électroniques – protocole SMTP (*Simple Mail Transfer Protocol*) –, établir une connexion à distance (protocole Telnet ou SSH) ou consulter un site Web touristique – protocole HTTP (*HyperText Transfer Protocol*).

Cette adresse IP, qui est composée de quatre octets en IPV4 et de 16 octets en IPV6 (en cours de déploiement), et qui sert de base à toute communication informatique, est unique dans le monde (pour toutes les adresses « publiques »). En décimal pour IPV4, elle est représentée par quatre nombres séparés par des points. L'adresse 134.157.15.2 en est un exemple. Dans une communication entre deux ordinateurs, chacun doit connaître l'adresse IP de l'autre. Ces adresses numériques sont principalement utilisées, sous leur forme binaire, lors de ce dialogue.

Pour un utilisateur, retenir les « numéros IP » des différents serveurs qu'il sollicite régulièrement peut vite devenir fastidieux. Afin de faciliter cette mémorisation, on « colle » généralement un nom à chaque adresse IP, par exemple *www.mit.edu* pour l'adresse *104.94.4.78*. Ces « adresses nommées » sont plus simples à retenir, et elles rappellent souvent une organisation géographique ou par service. Ainsi, l'adresse du portail d'Amazon en France est : www.amazon.fr. (*www* représente le service Web, *amazon* le nom du site et *fr* la France). Si l'usage de ces noms facilite le travail des utilisateurs, cela n'aide en rien les ordinateurs, qui n'utilisent que les numéros IP. Une table de correspondance entre les noms et les adresses IP est alors nécessaire pour permettre à l'ordinateur initiant la communication de trouver le numéro de son correspondant à partir du nom fourni. C'est le rôle du fichier `/etc/hosts` dans le système UNIX. Ce fichier texte fourni, à chaque ligne, la correspondance entre un numéro IP et un nom d'ordinateur (voir listing 2.1).

LISTING 2.1

```
#### Fichier  /etc/hosts ####
127.0.0.1          localhost      # NE PAS CHANGER
134.157.15.1       serv1
134.157.15.2       serv2      serv2.crip.univ.fr       serveur2
134.157.117.157    mangue     mangue.crip.univ.fr
```

La première ligne de ce fichier contient toujours l'adresse « privée » de l'ordinateur. Cette adresse est toujours la même et chaque ordinateur la possède. Elle permet, notamment, de tester le fonctionnement des couches réseaux internes au système. Les autres adresses montrent qu'il est possible de donner un ou plusieurs noms ou *alias* à une adresse IP. Ce fichier ne peut cependant pas contenir le nom de tous les ordinateurs du réseau Internet. Sa portée est généralement limitée aux machines géographiquement proches ou souvent utilisées. C'est le protocole DNS (*Domain Name System*) qui a la charge d'établir la correspondance entre un nom de machine et son adresse IP, quel que soit l'endroit où se trouve cet ordinateur. Avant l'apparition de ce DNS, un organisme, le NIC (*Network Information Center*), était chargé de référencer les adresses principales. Mais ce système est vite devenu ingérable. Pour plus d'informations, consulter le RFC (*Request For Comments*) 952.

Afin que tout ordinateur ou poste de travail puisse retrouver l'adresse IP attribuée à un nom (ou l'inverse), une hiérarchie de serveurs DNS a été constituée. Chaque serveur gère l'ensemble des noms et des adresses IP des ordinateurs se trouvant dans son domaine. Par exemple, le serveur DNS du domaine crip.univ.fr gère tous les ordinateurs dont le nom est du type xxxx.crip.univ.fr. Quand un ordinateur interroge son serveur DNS pour avoir l'adresse IP d'une machine distante, celui-ci retourne l'information si la machine en question est dans son domaine. Dans le cas contraire, le serveur DNS interroge le serveur DNS supérieur, qui va propager cette demande aux autres afin d'atteindre celui qui gère la machine recherchée. Ce dernier retourne alors l'adresse IP, qui arrive, par le parcours inverse, à l'ordinateur qui est à l'origine de la demande. Une fois le nom résolu, la communication entre l'ordinateur et la machine distante s'établit à partir des adresses IP.

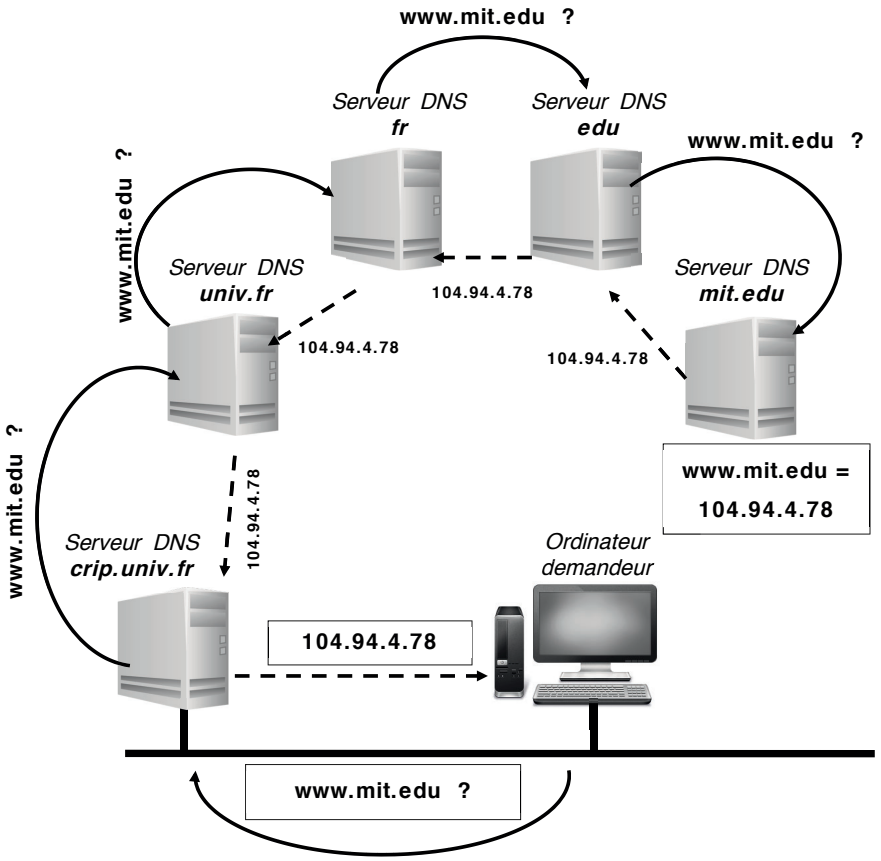


Figure 2.5

Hiérarchie de serveurs DNS.

La figure 2.5 présente le fonctionnement de cette hiérarchie de serveurs DNS, appliquée à la résolution du nom www.mit.edu à partir d'un ordinateur qui se trouve dans le domaine crip.univ.fr. Cette recherche est initiée lors de la saisie de l'URL

(*Uniform Resource Locator*) *http://www.mit.edu* dans le champ adresse d'un navigateur.

Les commandes UNIX `host` ou `nslookup` (ou `dig`, qui remplace cette commande dans de nombreuses distributions) permettent d'effectuer cette interrogation :

```
$ host www.mit.edu
www.mit.edu is an alias for www.mit.edu.edgekey.net.
www.mit.edu.edgekey.net is an alias for e9566.dscb.akamaiedge.net.
e9566.dscb.akamaiedge.net has address 104.94.4.78
```

Le nom officiel, l'alias (ou pseudonyme) ainsi que l'adresse IP 104.94.4.78 du serveur www.mit.edu sont fournis par la commande `host`. La commande `nslookup` donne une information similaire, à laquelle s'ajoutent le nom et l'adresse IP du serveur DNS initial qui a été sollicité dans cette résolution de nom :

```
$ nslookup www.mit.edu
Server:     serv3.crip.univ.fr
Address:    serv3.crip.univ.fr#53

Non-authoritative answer:
www.mit.edu canonical name = www.mit.edu.edgekey.net.
www.mit.edu.edgekey.net canonical name = e9566.dscb.akamaiedge.net.
Name:  e9566.dscb.akamaiedge.net
Address: 104.94.4.78
```

ATTENTION

Seule l'adresse IP est employée dans la communication entre ordinateurs. Le nom « collé » à cette adresse IP sert uniquement à faciliter le travail des utilisateurs en leur proposant des noms à la place de numéros. Avant toute communication entre deux ordinateurs, il faut d'abord retrouver l'adresse IP correspondant au nom fourni par l'utilisateur : c'est le rôle des serveurs DNS. Cette phase se nomme la *résolution du nom*. La communication commence une fois que l'adresse est trouvée.

Si l'un des serveurs DNS interrogé est en panne, l'adresse IP ne sera pas retournée et la communication ne pourra s'établir, même si le serveur à atteindre est opérationnel. Si un utilisateur connaît l'adresse IP du serveur qu'il veut atteindre, il peut directement l'employer à la place du nom, ce qui permet de commencer la communication immédiatement, sans passer par la phase de résolution du nom. Dans ce cas, les serveurs DNS ne sont plus utilisés.

À NOTER

Souvent, le fichier `/etc/hosts` est maintenu à jour de la même manière que les fichiers associés à la gestion du DNS. Ainsi, en cas de panne du service DNS, ce fichier local prend le relais pour les adresses qu'il contient (moyennant quelques manipulations systèmes simples).

2.3. Services réseaux et modes d'accès

Les adresses IP servent de support aux services réseaux, tout comme les adresses postales servent à d'autres services de la vie courante. De la même manière que les

sociétés de vente par correspondance utilisent l'adresse postale pour livrer leurs commandes, le protocole de transfert de fichier FTP (*File Transfer Protocol*) exploite l'adressage IP pour livrer des fichiers « à domicile ». La plupart de ces services sont basés sur le modèle client/serveur (ftp, http, smtp, …), mais l'un d'entre eux fonctionne différemment : il s'agit de Telnet (ou ssh).

Le modèle client/serveur

Dans ce mode de fonctionnement, le service est géré par deux logiciels. Un premier logiciel, le *client*, se trouve sur le poste de travail (micro-ordinateur). Il communique avec un second, le *serveur*, qui fait appel au même protocole et qui s'exécute sur l'ordinateur distant. C'est le modèle utilisé pour le Web. Un logiciel client de type navigateur (Firefox, Internet Explorer, Safari, …) interroge à distance un serveur Web. Dans le monde UNIX, ce serveur logiciel porte le nom de *httpd*. Le « d » final est mis pour « démon » (*daemon*). Ainsi, le serveur ftp se nomme *ftpd*, et le serveur courrier (protocole smtp) se nomme *smtpd*.

Ce modèle ne génère pas de connexion directe de l'utilisateur sur le serveur, mais une communication entre un logiciel client local et un logiciel serveur distant. Tout ce qui est entré par le biais du clavier ou de la souris est interprété sur le poste de travail. Le logiciel client (par exemple un navigateur), qui s'exécute en local, communique ensuite avec le serveur en utilisant la syntaxe du protocole, selon les directives (clics ou commandes) fournies par l'utilisateur (voir figure 2.6).

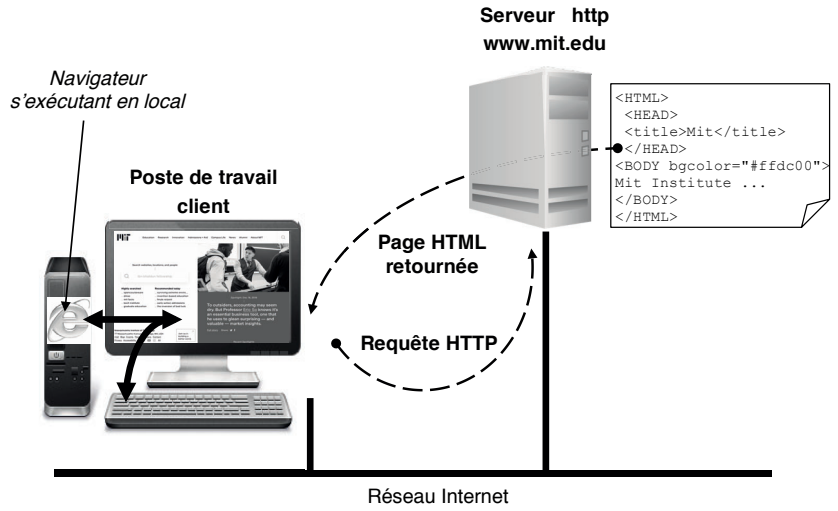


Figure 2.6
Le modèle client/serveur, http.

L'émulation de terminal : Telnet ou ssh

Le protocole Telnet (ou ssh) fonctionne différemment. Si la communication est bien établie initialement entre un client (`putty.exe, …`) et un serveur (`telnetd` ou

`sshd`), le reste de la communication se déroule autrement. Dans ce modèle, tout ce qui est saisi au clavier est directement transmis à l'ordinateur avec lequel la communication est établie, sans interprétation locale. C'est l'ordinateur distant qui exécute les commandes. De même, tout ce qui est envoyé par l'ordinateur distant est affiché directement sur l'écran du poste de travail (voir figure 2.7).

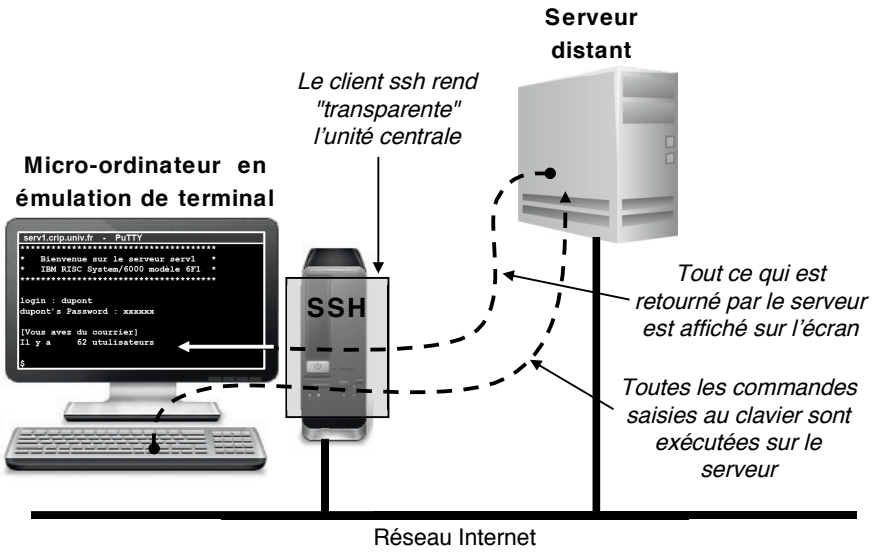


Figure 2.7
L'émulation de terminal : Telnet ou ssh.

Les protocoles Telnet et ssh sont largement utilisés sur les micro-ordinateurs. Ce sont des moyens d'accès privilégiés aux serveurs UNIX. Toutefois, il existe une autre méthode : le terminal X.

Le terminal X

Le terminal X est avant tout un terminal. Il possède une petite unité centrale, mais celle-ci ne contient aucun disque dur, contrairement aux micro-ordinateurs. Il va chercher son système d'exploitation sur un serveur distant lors de sa phase de démarrage. Une fois opérationnel, il se comporte comme un terminal. Deux différences importantes, cependant : il utilise le protocole graphique X-Window, qui permet le multifenêtrage et le déport d'affichage graphique lors de l'utilisation de logiciels graphiques hébergés sur le serveur distant, et il est capable de communiquer simultanément sur le réseau avec plusieurs serveurs.

Ce type de terminal, très largement utilisé par le passé, a progressivement disparu. Plus exactement, il a évolué en s'adaptant aux nouveaux systèmes d'exploitation. Il est devenu un « client léger » en embarquant des systèmes comme Windows ou Linux. Mais en se spécialisant, il a souvent perdu sa capacité à communiquer simultanément avec plusieurs serveurs.

Résumé

Le système UNIX a été conçu pour accueillir plusieurs connexions simultanément. Chaque utilisateur travaille à partir d'un poste distant qui joue le rôle d'interface de saisie et d'affichage, et qui communique avec l'ordinateur central. On appelle ce poste « passif » un *terminal*. La communication entre le terminal et le serveur emploie un langage particulier, qui prend la forme d'une suite de caractères de contrôles, la « suite d'échappement ». Ces terminaux sont aujourd'hui remplacés par des micro-ordinateurs qui communiquent avec le serveur distant *via* le réseau Internet. Un logiciel particulier, l'*émulateur de terminal*, les transforme en périphérique de communication.

Le recours au réseau permet de se connecter à partir du même poste sur plusieurs serveurs différents. Cela est possible avec des terminaux X, qui gèrent le multifenêtrage, en plus du réseau, mais également avec des PC émulés. Cette souplesse de connexion, qui n'existait pas quand les terminaux étaient connectés en direct sur le port série de l'ordinateur, exige quelques connaissances réseau. Le poste doit être paramétré avec l'adresse IP, l'adresse du DNS, etc., ce qui n'est pas le cas d'une connexion directe. Ce mode de connexion peut échouer si des dysfonctionnements du réseau apparaissent.

Questions de révisions

1. Qu'est-ce qu'un terminal ?
2. Qu'est-ce qu'un émulateur de terminal ?
3. Quel est le rôle d'un serveur DNS ?
4. Quel fichier UNIX peut jouer le rôle d'un « embryon » de service DNS ?
5. Dans les services réseaux, quelle est la différence entre le modèle « client/serveur » et le modèle « émulation de terminal » ?
6. Qu'est-ce qu'une URL ?

Exercices

1. Affichez l'adresse IP du serveur *www.ibm.com*.
2. Donnez le nom des protocoles utilisés pour le transfert de fichiers et le transfert des courriers électroniques.
3. Donnez le nom du terminal de la gamme DEC qui sert de référence aux émulateurs.
4. Que signifie le terme *IP* ?
5. Comment se nomme le protocole graphique utilisé sous UNIX ?
6. Que signifie le terme *http* ?

3

La première session

PLAN

1. La connexion
2. La déconnexion
3. L'environnement standard de travail
4. Les utilisateurs connectés
5. L'identification d'un utilisateur

OBJECTIFS

- S'initier à une ouverture de session et à quelques commandes élémentaires
- Comprendre la notion d'identification
- Définir son terminal de travail
- Modifier les paramètres de son compte utilisateur

Dans le chapitre précédent, nous avons présenté la méthode d'accès au serveur UNIX ou Linux. Dans celui-ci, nous étudions comment se connecter, se déconnecter et exécuter des commandes. Nous nous appuierons sur les connaissances acquises précédemment, comme le protocole Telnet ou la notion d'adressage IP.

1. La connexion

La connexion sur un système UNIX utilise un système d'authentification basé sur un login et un mot de passe. Le système demande tout d'abord le login – c'est la phase d'identification –, puis le mot de passe – c'est la phase d'authentification. Une fois qu'il est reconnu, l'utilisateur se retrouve connecté : une session est ouverte. L'invite de saisie de l'identification change selon l'interface, qui peut être graphique ou texte. Voyons tout d'abord ces interfaces.

9782340-038004_001_828.indd 45 01/04/2020 16:34

1.1. Les interfaces

Il existe deux interfaces, ou *modes de connexions* :

- l'*interface graphique*, également appelée *mode graphique* ;
- l'*interface texte*, également appelée *mode commande*.

Ces deux interfaces sont présentes dans l'environnement UNIX particulier aux « gros ordinateurs », et dans l'environnement Linux dédié aux micro-ordinateurs. Mais il est plus courant de trouver l'interface texte sur les serveurs administrés sous UNIX et l'interface graphique pour Linux.

Le mode graphique

Le mode graphique repose sur le protocole X-Window, ou X11. Ce protocole est employé lors d'une ouverture de session à partir d'un terminal X ou d'un émulateur X. Les serveurs UNIX proposent généralement un bureau graphique s'appuyant sur X-Window : il s'agit de CDE (*Common Desktop Environment*). Ce bureau normalisé présente la même interface sur l'ensemble des serveurs qui l'implémentent. Il propose des fonctionnalités et des utilitaires intégrés dans une barre de menus, ainsi qu'un bureau virtuel, disponible sous la forme de quatre « écrans » différents, accessibles d'un simple clic dans la barre. Vous pouvez modifier cette interface afin de l'adapter à vos besoins (ajout d'outils dans la barre de menus, choix du nombre de bureaux virtuels, choix de la police…).

Linux dispose de ses propres bureaux graphiques. Les deux principaux se nomment *KDE* (*K Desktop Environnement*) et *GNOME* (*GNU Network Object Model Environment*). KDE est une « adaptation » de CDE à l'environnement Linux. Il est très utilisé et présente le double avantage d'être très proche de l'ergonomie de CDE et du bureau Windows. Le fait de choisir KDE ou GNOME n'a aucune incidence sur le fonctionnement des applications, qui sont en majorité indépendantes du type de bureau. Ce choix est principalement dicté par l'appréciation de la présentation graphique et des outils associés à chaque interface. Quel que soit le bureau utilisé (CDE, KDE ou GNOME), une bannière graphique nous invite à saisir le nom de login et le mot de passe (voir figures 3.1 et 3.2).

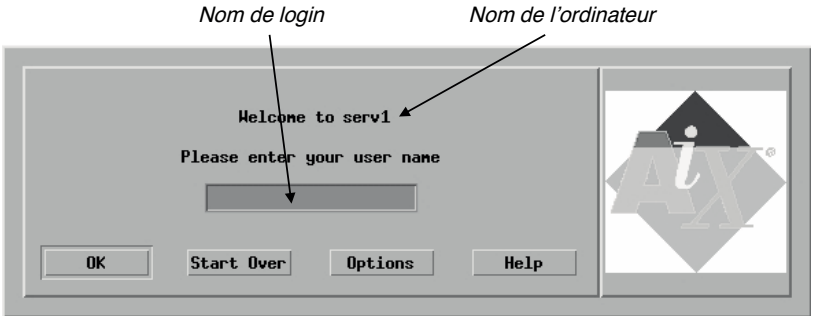


Figure 3.1
Bannière graphique de connexion sous AIX avec CDE.

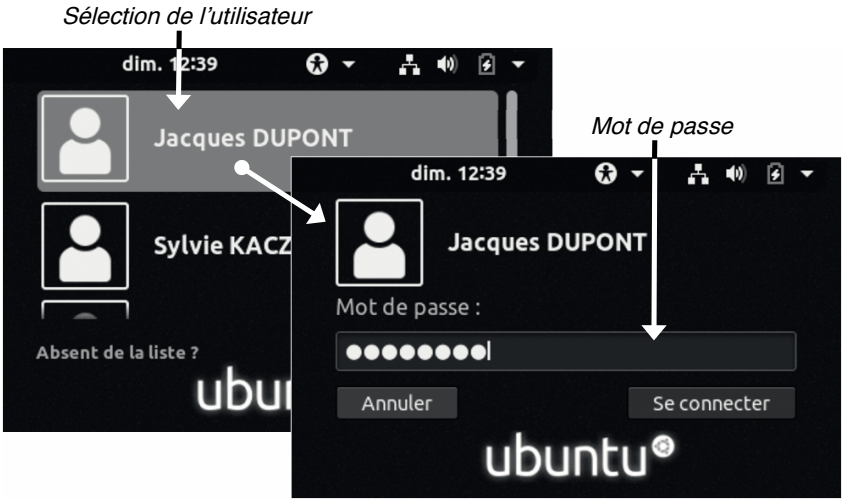


Figure 3.2

Bannière graphique de connexion sous Linux avec GNOME.

Une fois que l'on est connecté, le bureau s'affiche à l'écran. L'ouverture d'une « fenêtre terminal » donne accès aux commandes UNIX. La figure 3.3 présente le bureau CDE et le menu d'ouverture d'une fenêtre terminal. La première étape fait apparaître le menu, la seconde sélectionne l'ouverture du « terminal ».

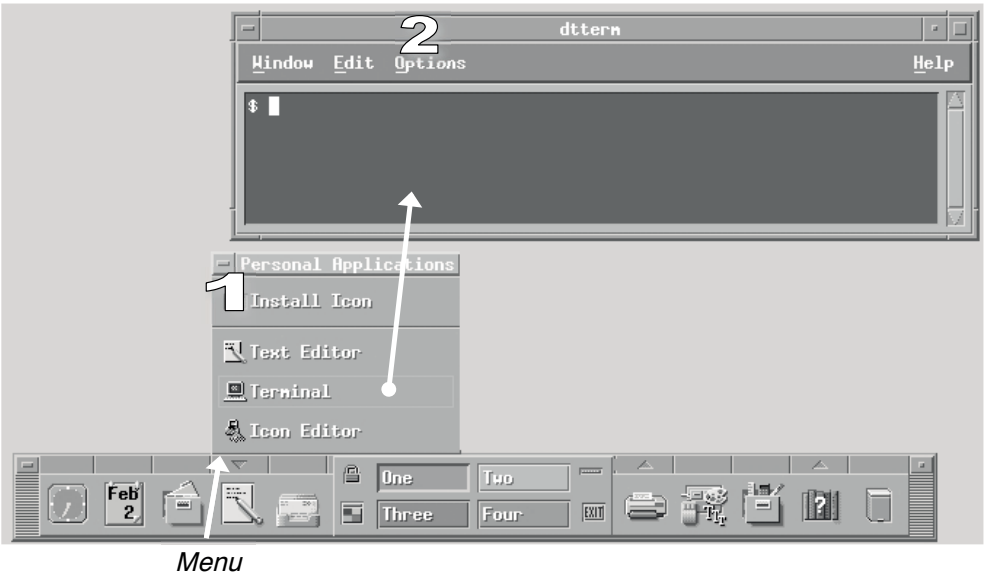


Figure 3.3

Ouverture d'une fenêtre terminal sous CDE.

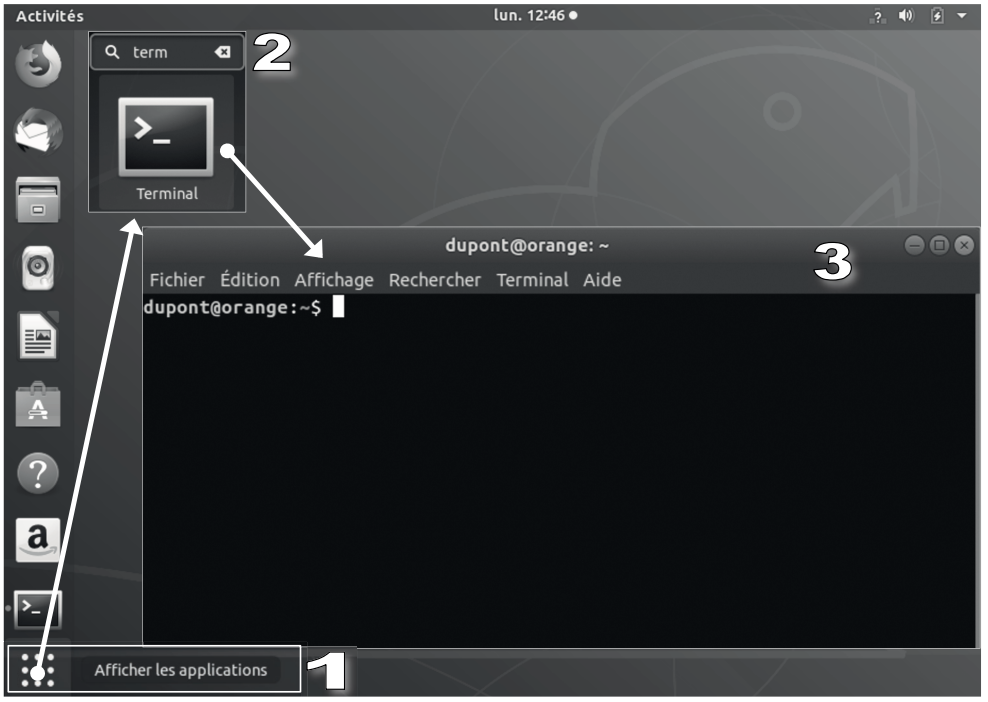


Figure 3.4

Ouverture d'une fenêtre « Terminal » sous GNOME.

Dans l'environnement graphique de Linux, il est possible d'ouvrir une fenêtre *xterm* émulant un terminal X ou une fenêtre « Terminal ». Avec le bureau GNOME, l'ouverture d'une fenêtre « Terminal » s'obtient à partir du « Menu des applications » (étape 1), puis la recherche et la sélection de « Terminal » (étape 2), la fenêtre principale apparaît (étape 3) comme cela est présenté à la figure 3.4.

Le mode commande

L'interface texte est plus simple, et est souvent privilégiée par les administrateurs. On l'obtient dès l'allumage du terminal, dans le cas d'une connexion série (connexion directe), ou par le lancement de la commande `telnet` sur un micro-ordinateur. Avec cette interface, une session UNIX peut être ouverte à partir de n'importe quel poste de travail.

La figure 3.5 présente les trois menus ou fenêtres successifs de l'exécution de la commande telnet sous Windows. Dans le menu de recherche, sélectionnez « cmd » (étape 1), puis dans le menu, choisissez « Invite de commandes » (étape 2). Dans la nouvelle fenêtre saisissez la commande de connexion vers l'ordinateur UNIX comme par exemple : `telnet ananas.crip.univ.fr` (étape 3) suivie de l'identifiant et du mot de passe.

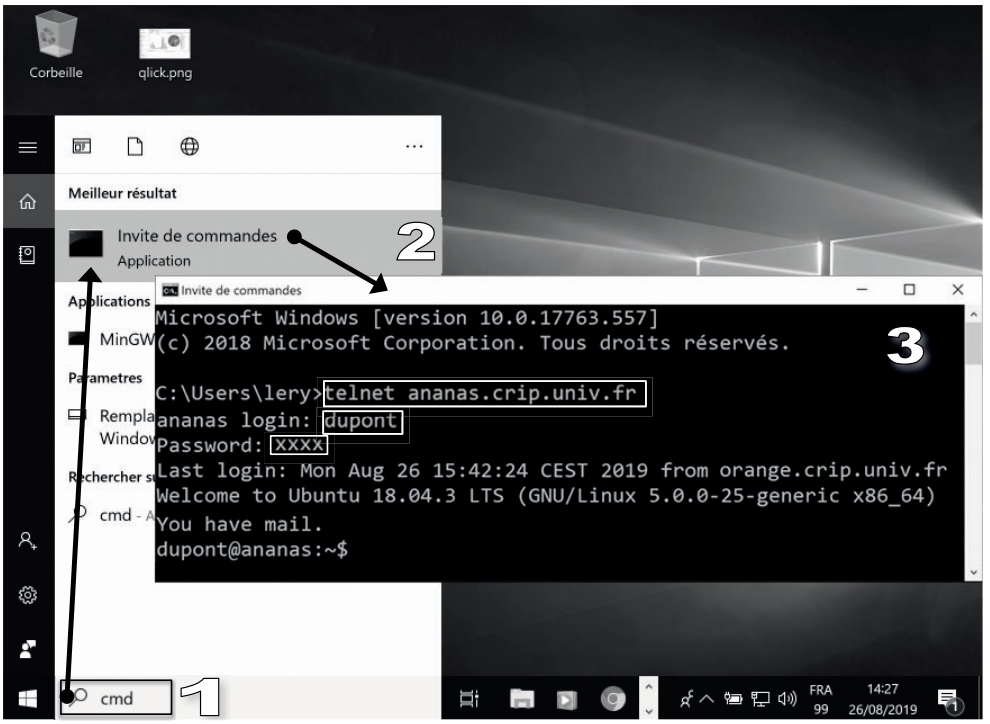


Figure 3.5

Lancement de la commande *telnet* sous Windows.

À NOTER

Il est aussi possible d'utiliser le logiciel libre `putty.exe`. Il suffit de le télécharger sur Internet à l'adresse **http://www.putty.org**. Le lancement de la commande `putty` est identique à celui de `telnet` sous Windows ; en outre, et ce n'est pas négligeable, ce logiciel offre la possibilité de sécuriser la connexion *via* ssh.

Une fois que la connexion est établie, le serveur vous invite à taper votre identifiant et votre mot de passe.

```
$ telnet ananas.crip.univ.fr
Trying 10.157.15.114...
Connected to ananas.crip.univ.fr.
Escape character is '^]'.
Ubuntu 18.04.3 LTS
ananas login: dupont
Password: xxxx
Last login:Mon Aug 26 15:45:34 CEST 2019 from orange on pts/0
Welcome to Ubuntu 18.04.3 LTS (GNU/Linux 5.0.0-25-generic x86_64)
22 paquets peuvent être mis à jour.
You have mail.
dupont@ananas:~$
```

1.2. L'authentification

Pour s'authentifier sur un système UNIX ou Linux, il faut disposer d'un *identifiant* (*login* en anglais) et d'un *mot de passe* (*password* en anglais). Cette méthode est très répandue et les systèmes sans authentification (comme Windows 95 ou 98) ont totalement disparu. Si la méthode est devenue courante, les contraintes et les règles de sécurité qui y sont rattachées sont moins souvent connues des utilisateurs.

Le nom de login

Le nom de login permet à l'utilisateur de s'identifier. Lors de l'identification, le système recherche, parmi la liste des utilisateurs, le nom saisi. Ce nom peut être choisi librement (sauf politique particulière imposée par l'administrateur et problèmes d'homonymie), mais il est soumis à certaines contraintes.

La première des contraintes est la limitation de la taille du nom de login à *8 caractères*. Cette limitation est historique et elle est encore d'actualité sur certains anciens systèmes UNIX. Elle est peu connue des utilisateurs, et parfois même des administrateurs. En effet, elle n'est pas flagrante car la saisie d'un nom supérieur à 8 caractères ne pose aucun problème. Dans ce cas, seuls les 8 premiers caractères sont utilisés par le système. Ainsi, deux utilisateurs possédant respectivement les noms de login `benedette` et `benedetto` (sur 9 caractères) seraient identifiés comme un seul et même utilisateur, `benedett` (sur 8 caractères). Cette contrainte a disparu sur les versions récentes des systèmes AIX et Linux.

La seconde contrainte est la différence entre les *majuscules* et les *minuscules*. UNIX considère les majuscules et les minuscules comme des caractères différents. C'est effectivement ce que définit la table ASCII (*American Standard Code for Information and Interchange*). Une saisie du nom de login en majuscules provoque une identification sous un « autre nom ». Voici une exécution sous AIX :

```
login: DUPONT
3004-033 YOU LOGGED IN USING ALL UPPERCASE CHARACTERS. IF YOUR
     WORKSTATION ALSO SUPPORTS LOWERCASE CHARACTERS, LOG OFF,
     THEN LOG IN AGAIN USING LOWERCASE CHARACTERS.
\D\U\P\O\N\T'S \PASSWORD: xxxxx
3004-007 \YOU ENTERED AN INVALID LOGIN NAME OR PASSWORD.
```

Voici la même exécution sous Linux avec le message d'erreur « Login incorrect » :

```
login: DUPONT
Password: xxxxx
Login incorrect
```

ATTENTION

Certains systèmes UNIX limitent encore le nom de login à 8 caractères. Vérifiez si votre système présente cette contrainte, et n'utilisez que des minuscules. De plus, certains caractères « spéciaux » sont à proscrire, comme l'espace, l'astérisque (*), le dollar ($), le point d'interrogation (?), etc., ainsi que les caractères accentués. Employez essentiellement les lettres alphabétiques non accentuées et les chiffres.