Septième partie

Appendices

Bibliographie

[3Com501] 3Com Corporation, **Etherlink (3C501) Adapter Technical Reference**, Manual Part 6405-00, 1988, 21 p.

[ABR–70] Abramson, N., *The Aloha System-Another Alternative for Computer Communications*, Compte rendu de la conférence **Fall Joint Computer**, conférence AFIPS, p.37, 1970.

[ABR–85] Abramson, N., *Development of the Alohanet*, **IEEE Transactions on Information Theory**, vol.IT–31, March 1985, pp.119–123.

[BAR–60] Baran, Paul, *Reliable Digital Communications Systems Using Unreliable Network Repeater Nodes*, **RAND Corporation Mathematics Division Report** n^o P-1995, 27 mai 1960.

[BAR–64] Baran, Paul, *On Distributed Communications Networks*, **IEEE Transactions on Information Theory**, 1er mars 1964.

[BEC–96] Beck, Michael & Böhme, Harald & Dziadzka, Mirko & Kunitz, Ulrich & Magnus, Robert & Verworner, Dirk, **Linux-Kernel-Programmierung**, Addison-Wesley (Deutschland); traduction anglaise **Linux Kernel Internals**, Addison-Wesley, 1996; second edition, 1998, XVI + 480 p. + CD-ROM; third edition, Addison-Wesley, 2002, XIV + 471 p. + CD-ROM.

[BOV–01] Bovet, Daniel & Cesati, Marco, **Understanding the Linux Kernel: from I/O ports to process management**, O'Reilly, 2001, XVI + 684 p.; traduction française **Le noyau Linux**, O'Reilly, 2001, XVI + 673 p.

[Le plus détaillé des livres sur l'implémentation de Linux avant [CEG–03], le premier à aborder les points concernant l'architecture du microprocesseur (en l'occurrence le `80x86` d'Intel). Le choix du dernier noyau de l'époque, le noyau 2.2, ne permet pas de commenter tous les points essentiels d'un système d'exploitation, ce qui est dommage vu la qualité de ce livre.]

[CAR–98] Card, Rémy & Dumas, Éric & Mével, Franck, **Programmation Linux 2.0 : API système et fonctionnement du noyau**, Eyrolles, 1998, XIII + 520 p. + CD-ROM, ISBN 2-212-08932-5; traduction anglaise **The Linux Kernel Book**, Wiley, 1998.

[Comme son nom l'indique, son but est d'aider le programmeur Linux. Il donne cependant des notes sur l'implémentation, en décrivant surtout les structures utilisées (pour le noyau 2.0).]

[CEG–03] Cégielski, Patrick, **Conception des systèmes d'exploitation : le cas Linux**, Eyrolles, 2003, XIII + 595 p., ISBN 2-212-11360-9; deuxième édition, XIII + 680 p., septembre 2004.

[Prend comme exemple de système d'exploitation le noyau Linux 0.01 dont il commente le code complet.]

[CK–74] Cerf, Vincent & Kahn, R., *A Protocol for Packet Network Interconnection*, **IEEE Transactions on Communications Technology**, vol. COM-22, no. 5, 1974, pp. 627–641.

[C-P–02] Crowcroft, Jon & Phillips, Iain, **TCP/IP and Linux Protocol Implementation**, Wiley, 2002, L + 925 p., ISBN 0-471-40882-4.

[Extrait de code ordonné avec assez peu de commentaires.]

[DAV–00] Davis, Martin, **The Universal Computer**, Norton, 2000, XII + 257 p.

[GRA–00] Gray, Warren W., **Linux Socket Programming By Example**, QUE, 2000, XV + 558 p., ISBN 0-7897-2241-0.

[HER–00] Herrin, David, **Linux IP Networking: A Guide to the Implementation and Modification of the Linux Protocol Stack**, disponible en ligne :

`http://kernelnewbies.org/documents/ipnetwoorking/linuxipnetworking.html`

[HL–96] Hafner, Katie & Lyon, Matthew, **Where Wizards stay up late**, Simon & Schuster, 1996; traduction française **Les sorciers du Net : les origines de l'Internet**, Calmann-Lévy, 1999, 347 p.

[HUU–03] Huurdeman, Anton, **The Worldwide History of Telecommunications**, Wiley, 2003, XX+638 p.

[Très bon travail de synthèse sur les télécommunications avant l'apparition des réseaux informatiques. Ne comprend pas que le système de commutation est intrinsèquement différent pour les réseaux informatique et que ces derniers vont détrôner les premiers (avec la technique de la voix sur IP).]

[IEEE-802] **IEEE Standards for Local and Metropolitan Area Networks: Overview and Architecture**, 1990, 31 p., ISBN 1-55937-052-1. Téléchargeable à :

`http://standards.ieee.org/catalog/olis/802-1990.pdf`

[IEEE-802.2] **IEEE Standards for Local and Metropolitan Area Networks. Part 2: Logical Link Control**, 1998, 253 p., aussi ISO/IEC 8802.2:1998. Téléchargeable à :

`http://grouper.ieee.org/groups/scc32/dsrc/ip/ip_images/802.2-1998.pdf`

[IEEE-802.3] **IEEE Standards for Local and Metropolitan Area Networks. Part 3: Carrier sense multiple access with collision detection (CSMA/CD) access method and physical layer specifications**, 2002, 1 538 p. Téléchargeable à :

`http://standards.ieee.org/getieee802/download/802.3-2002.pdf`

[K-D–01] Kirch, Olaf & Dawson, Terry, **Administration réseau sous Linux**, O'Reilly, seconde édition, janvier 2001, 544 p., ISBN : 2-84177-125-3.

[K-R–01] Kurose, James F. & Ross, Keith W., **Computer Networking: A Top Down Approach Featuring the Internet**, Addison-Weslay, 2001, XXIV + 712 p., ISBN 0-201-47711-4, third edition, 2004; traduction française de la deuxième édition **Analyse structurée des réseaux**, Pearson, 2003, 900 p, ISBN: 2-7440-7000-9.

[ISO 7498-1] **ISO/IEC 7498 Part 1: The Basic Model**

[La seconde version, celle de 1994, est disponible auprès du site web de ISO :

`http://www.iso.org/iso/en/CatalogueDetailPage.CatalogueDetail?CSNUMBER=20269` en anglais ou en français, sous forme électronique ou papier, au prix de 160 CHF. Une version texte antérieure peut être chargée gratuitement sur le site web de l'ACM :

`http://www.acm.org/sigs/sigcomm/standards/iso_stds/OSI_MODEL/ISO_IEC_7498-1.TXT`

]

[ISO 7498-2] **ISO/IEC 7498 Part 2: Security Architecture**

[La version de 1989 est disponible auprès du site web de ISO :

`http://www.iso.org/iso/en/CatalogueDetailPage.CatalogueDetail?CSNUMBER=14256`

en anglais ou en français, sous forme électronique ou papier, au prix de 116 CHF.]

[ISO 7498-3] **ISO/IEC 7498 Part 3: Naming and Addressing**

[La version de 1997 est disponible auprès du site web de ISO :

`http://www.iso.org/iso/en/CatalogueDetailPage.CatalogueDetail?CSNUMBER=25022`

en anglais ou en français, sous forme électronique ou papier, au prix de 97 CHF. Une version texte antérieure peut être chargée gratuitement sur le site web de l'ACM :

`http://www.acm.org/sigs/sigcomm/standards/iso_stds/OSI_MODEL/ISO_IEC_7498-3.TXT`

]

[ISO 7498-4] **ISO/IEC 7498 Part 4: Management Framework**

[La version de 1989 est disponible auprès du site web de ISO :

`http://www.iso.org/iso/en/CatalogueDetailPage.CatalogueDetail?CSNUMBER=14258`

en anglais ou en français, sous forme électronique ou papier, au prix de 61 CHF.]

[KLE–61] Kleinrock, Leonard, *Information Flow in Large Communication Networks*, **RLE Quaterly Progress Reports**, juillet 1961.

[KLE–86] Kleiman, S. *Vnodes: An Architecture for Multiple File System Types in Sun Unix*, **Proccedings of the Summer USENIX Conference**, June 1986, pp. 260–269.

[KLE–64] Kleinrock, Leonard, **Communication Nets: Stochastic Message Flow and Delay**, McGraw-Hill, 1964.

[LEF–89] Leffler, Samuel J. & McKusick, Marshall Kirk & Karels, Michael J. & Quarterman, John S., **The Design and Implementation of the 4.3 BSD UNIX Operating System**, Addison-Wesley, 1989 (reprinted with corrections on October, 1990), 471 p., ISBN 0-201-06196-1.

[MAN–01] Mancill, Tony, **Linux Routers: A Primer for Network Administrators**, Prentice Hall, 2001.

[McK–96] McKusick, Keith Bostic, Marshall Kirk & Karels, Michael J. & Quarterman, John S., **The Design and Implementation of the 4.4 BSD UNIX Operating System**, Addison-Wesley, 1996, ISBN 0-201-54979-4; traduction française, Vuibert, 1997, 576 p, 2-84180-142-X.

[M-R–76] Metcalfe, R. M. et Boggs, D. R., *Ethernet: Distributed Packet Switching for Local Computer Networks*, **Communications of the Association for Computing Machinery**, vol. 19, July 1976, pp.395–404.

[PAR–96] Parker, Timothy, **Teach Yourself TCP/IP in 14 days**, Sams, 1996, ISBN 0-672-30885-1; traduction française **TCP/IP**, Simon & Schuster Macmillan, 1996, VII + 450 p.

[PUJ–04] Pujolle, Guy, **Réseaux**, Eyrolles, cinquième édition, 2004, 1094 pages, ISBN : 2-212-11437-0.

[RFC 1] Crocker, S., **Host Software**, 7 April 1969, 11 p., RFC 1.

[RFC 768] Postel, Jon, **User Datagram Protocol**, IETF, 29 August 1980, RFC 768, 3 p. Traduction française par F.G. Fremaux.

[RFC 791] Postel, Jon, **Internet Protocol: Darpa Internet Program Protocol Specification**, September 1981, 45 p., RFC 791. Traduction française par F.G. Fremaux.

[RFC 792] Postel, Jon, **Internet Control Message Protocol**, September 1981, 21 p., RFC 792. Traduction française.

[RFC 793] Postel, Jon, **Transmission Control Protocol**, September 1981, 85 p., RFC 793. Traduction française par F.G. Fremaux.

[RFC 826] Plumer, David C., **An Ethernet Address Resolution Protocol: Converting Network Protocol Addresses to 48.bit Ethernet Address for Transmission on Ethernet Hardware**, November 1982, RFC 826.

[RFC 1071] Braden, R., Borman, D. & Partridge, C., **Computing the Internet Checksum**, IETF, September 1988, 24 p., RFC 1071.

[RFC 1122] Braden, R., **Requirements for Internet Hosts – Communication Layers**, October 1989, 116 p., RFC 1122.

[RFC 1141] Mallory, T. et Kullberg, A., **Incremental Updating of the Internet Checksum**, IETF, January 1990, 2 p., RFC 1141.

[RFC 1310] Chapin, Lyman, **The Internet Standards Process**, March 1992, 23 p., RFC 1310.

[RFC 1518] Rekhter, Yakov et Li, Tony, **An Architecture for IP Address Allocation with CIDR**, IETF, September 1993, 27 p., RFC 1518.

[RFC 1519] Fuller, Vince, Li, Tony, Yu, Jessica & Varadhan, Kannan, **Classless Inter-Domain Routing (CIDR): an Address Assignment and Aggregation Strategy**, IETF, September 1993, 24 p., RFC 1519.

[RFC 1624] Rijsinghani, Anil, **Computation of the Internet Checksum via Incremental Update**, IETF, May 1994, 6 p., RFC 1624.

[RFC 1700] Reynolds, J & Postel, Jon, **Assigned Numbers**, October 1994, 230 p., RFC 1700.

[RFC 1812] Baker, Fred, **Requirements for IP Version 4 Routers**, June 1995, 175 p., RFC 1812.

[RFC 1889] Schulzrinne, H., Frederik, R. & Jacobson, V., **RTP: A Transport Protocol for Real-Time Applications**, January 1996, 75 p., RFC 1889.

[RFC 2292] Stevens, W. & Thomas, M., **Advanced Sockets API for IPv6**, February 1998, 67 p., RFC 2292.

[RFC 2474] Nichols, K. & Blake, S. & Baker, F. & Black, D. **Definition of the Differentiated Services Field (DS Field) in the IPv4 and IPv6 Headers**, December 1998, 20 p., RFC 2474.

[SAT–00] Satchell, Stephen T., **Linux IP Stacks Commentary**, Coriolis Open Press, 2000.

[STE–90] Stevens, W. Richard, **Unix Network Programming**, Prentice-Hall, 1990, XI + 772 p., second edition in two volumes in 1999, third edition in two volumes.

[TAN–81] Tanenbaum, Andrew, **Computer Networks**, Prentice-Hall, 1981, second edition, 1988, fourth edition, 2002; traduction française de la quatrième édition **Réseaux**, Pearson, 2003, XI + 908 p., ISBN : 2-7440-7001-7.

[TJ–97] Tischer, Michael & Jennrich, Bruno, **Internet Bible**, Data Becker, 1997; traduction française **La Bible Internet : expertise et programmation**, Micro Application, 1997, 1545 p. + CD-ROM, ISBN 2-7429-0762-9.

[UFF–87] Uffenbeck, John, **The 80x86 Family: Design, Programming, and Interfacing**, Prentice-Hall, 1987, 1998, third edition 2002, IX + 678 p. + CD-ROM.

[WPRMB–02] Wehrle, Klaus & Phlke, Frank & Ritter, Hartmut & Müller, Daniel & Bechler, Marc, **Linux Netzwerkarchitektur**, Addison-Wesley Verlag, 2002 ; traduction française **Architecture réseau Linux : conception et implémentation des protocoles réseau du noyau Linux**, Vuibert Informatique, 2003, XV + 726 p.

[Le but de l'ouvrage est de documenter l'API des réseaux de Linux pour implémenter de nouveaux protocoles et de nouveaux périphériques. Ce faisant, un éclairage fort intéressant sur la conception, sans entrer dans le détail du code, est fourni.]

Index

_SK_MEM_OVERHEAD, 412
_SK_MEM_PACKETS, 412
__BIG_ENDIAN, 227
__BIG_ENDIAN_BITFIELD, 227
__LINK_STATE_LINKWATCH_PENDING, 207
__LINK_STATE_NOCARRIER, 207
__LINK_STATE_PRESENT, 207
__LINK_STATE_RX_SCHED, 207
__LINK_STATE_SCHED, 207
__LINK_STATE_START, 207
__LINK_STATE_XOFF, 207
__LITTLE_ENDIAN, 227
__LITTLE_ENDIAN_BITFIELD, 227
___pskb_trim(), 195
__dev_alloc_skb(), 197
__dev_get_by_index(), 247
__dev_get_by_name(), 237
__dev_put(), 614
__dev_remove_pack(), 308
__in_dev_get(), 396
__init, 139
__initcall(), 140
__ip_evictor(), 635
__ip_route_output_key(), 572
__ip_select_ident(), 584
__kernel_caddr_t, 122
__kernel_clock_t, 122
__kernel_daddr_t, 122
__kernel_gid16_t, 122
__kernel_gid_32_t, 122
__kernel_gid_t, 122
__kernel_ino_t, 122
__kernel_ipc_pid_t, 122
__kernel_loff_t, 122
__kernel_mode_t, 122
__kernel_nlink_t, 122
__kernel_off_t, 122
__kernel_old_dev_t, 122
__kernel_old_gid_t, 122
__kernel_old_uid_t, 122
__kernel_pid_t, 122
__kernel_ptrdiff_t, 122
__kernel_size_t, 122
__kernel_ssize_t, 122
__kernel_suseconds_t, 122
__kernel_time_t, 122
__kernel_uid16_t, 122
__kernel_uid32_t, 122
__kernel_uid_t, 122
__kfree_skb(), 164
__lock_sock(), 418
__netdev_watchdog_up(), 285
__netif_rx_schedule(), 309
__netif_schedule(), 601
__pskb_pull(), 190
__pskb_pull_tail(), 191
__pskb_trim(), 195
__raise_softirq_irqoff(), 310
__release_sock(), 416
__s16, 123
__s32, 123
__s64, 123
__s8, 123
__sk_add_node(), 517
__sk_dst_reset(), 505
__sk_dst_set(), 553
__sk_head(), 401
__skb_append(), 179
__skb_dequeue(), 176
__skb_dequeue_tail(), 177
__skb_insert(), 178
__skb_linearize(), 592
__skb_pull(), 169
__skb_queue_head(), 174
__skb_queue_purge(), 178
__skb_queue_tail(), 175
__skb_trim(), 195
__skb_unlink(), 177
__sock_create(), 487
__sock_recvmsg(), 524
__sock_sendmsg(), 543

__u16, 123
__u32, 123
__u64, 123
__u8, 123
__udp_checksum_complete(), 438
10Base-F, 46
10Base2, 46
10Base5, 46
3C501.h
  l.54, 270
  l.61–70, 270
  l.72–79, 270
  l.80, 270
  l.82–88, 270
  l.90–92, 270
3Com, 33
3Com 501, 93
3c501.c
  l.1–100, 268
  l.139–201, 271
  l.203–325, 274
  l.327–361, 286
  l.363–387, 610
  l.390–494, 604
  l.496–686, 292
  l.548–633, 607
  l.635–673, 294
  l.689–754, 296
  l.756–786, 287
  l.788–816, 614
  l.818–833, 318
  l.885–889, 278
  l.891–898, 232
3c501.h
  l.102, 277
  l.103–107, 277
  l.18, 273
  l.20–24, 277
  l.26–38, 277
  l.41–53, 269
  l.56–59, 269
802/, 119

1822
  protocole, 16

abonné
  au téléphone, 4
accept(), 76
ACK, 39
acknowledge, 39
activité, 129
adaptateur réseau, 93
add_timer(), 128
adjacentes (machines), 35
adresse
  de sous-réseau, 106
  générique, 72
  Internet, 61
    par défaut, 73
  IP, 59, 61
    default, 326
  MAC, 100
  Unix, 72
AF_APPLETALK, 67, 442
AF_ASH, 442
AF_ATMPVC, 67, 442
AF_ATMSVC, 442
AF_AX25, 67, 442
AF_BLUETOOTH, 442
AF_BRIDGE, 442
AF_DECnet, 442
AF_ECONET, 442
AF_INET, 67, 442
af_inet.c
  l.1017–1122, 453
  l.1045–1049, 444
  l.1051–1064, 399
  l.1124, 453
  l.1139–1166, 457
  l.124–127, 456
  l.128, 453
  l.170–189, 545
  l.226–350, 495
  l.353–383, 484
  l.385–467, 511
  l.649–659, 544
  l.676–728, 502
  l.803–822, 480
  l.849–853, 443
  l.859–894, 448
  l.874–882, 447
  l.896, 456
  l.898–952, 452
  l.969–989, 449
  l.981–985, 398
  l.991–1012, 456
AF_INET6, 67, 442
AF_IPX, 67, 442
AF_IRDA, 442

AF_KEY, 442
AF_LLC, 442
AF_LOCAL, 67, 442
AF_MAX, 442
AF_NETBEUI, 442
AF_NETLINK, 67, 442
AF_NETROM, 442
AF_PACKET, 67, 442
AF_PPPOX, 442
AF_ROSE, 442
AF_ROUTE, 442
AF_SECURITY, 442
AF_SNA, 442
AF_UNIX, 67, 442
AF_UNSPEC, 442
AF_WANPIPE, 442
AF_X25, 67, 442
aio.c
    l.311–324, 523
aio.h
    l.46-78, 522
    l.81–94, 523
alloc_divert_blk(), 245
alloc_etherdev(), 262
alloc_netdev(), 263
alloc_skbuff(), 161
ALOHANET, 32
ancillary data, 87
anneau à jeton, 48
API
    réseau, v
appel système, 129
appletalk/, 119
application layer, 10
arch/, 118
architecture
    hybride, 11
    Internet, 55
    réseau, 7
    TCP/IP, 55
ARCNET, 33
ARP, 60
arp.c
    l.1235–1246, 456
    l.138–156, 588
arp_generic_ops, 588
arp_hh_ops, 588
arp_init(), 456
ARPA, 15
ARPANET, 15
arphdr (struct), 154
ARPHRD_CSLIP, 216
ARPHRD_ETHER, 216
ARPHRD_IEEE802, 216
ARPHRD_LOOPBACK, 216
ARPHRD_PPP, 216
ARPHRD_SLIP, 216
ARPHRD_VOID, 216
ASSERT_RTNL(), 243
AT&T, 14
ATM, 59
atm/, 119
atomic.h
    l.19–24, 133
    l.26–58, 134
    l.8–11, 133
atomic_add(), 134
atomic_dec(), 134
atomic_inc(), 134
ATOMIC_INIT(), 134
atomic_read(), 134
atomic_set(), 134
atomic_sub(), 134
atomic_t, 133
atomicité, 132
ax25.h, 73
ax25/, 119
AX_CMD, 269
AX_LOOP, 270
AX_OFF, 270
AX_RESET, 270
AX_RX, 270
AX_STATUS, 269
AX_SYS, 270
AX_XMIT, 270

backbone, 25, 29
backlog, 301
backplane, 30
BADCLASS(), 393
bande passante, 42
Baran, Paul, 6, 14
base de données
    de redirection, 327
BBN, 16
Bell Telephone Company, 4
Bell, Alexander Graham, 3
BH, 131
big_endian.h
    l.1–9, 227

bind(), 76
Biro, Ross, 144
bit
  d'arrêt, 23
  de début, 23
BITNET, 30
bitops.h, 133
BITS_PER_LONG, 123
Bluetooth, 25, 48
bluetooth/, 119
bottom half, 131
bps, 23
bridge/, 119
broadcast, 26, 100
BSD, 56
BUG(), 165
bug.h
  l.21–23, 168
  l.25–32, 166
  l.6–20, 165
BUG_ON(), 168
BUG_TRAP(), 186
bus, 29
bus à jeton, 48
BUS_ID_SIZE, 252
byteorder.h/asm-i386
  l.1–59, 227

câble
  de descente, 30
  Ethernet
    fin, 28
  transcepteur, 47
cache
  d'en-têtes matériel, 199
  de fragment, 632
  de routage, 363
cafteur, 525
canal, 42
  à accès aléatoire, 43
  à accès multiple, 43
  à contention, 42
  bruité, 35
  parfait, 35
CAP_NET_ADMIN, 446
CAP_NET_BIND_SERVICE, 446
CAP_NET_BROADCAST, 446
CAP_NET_RAW, 446
capability.h
  l.141–169, 446
carte
  Ethernet, 47
  réseau, 47, 93
central téléphonique, 4
Cerf, Vincent, 33, 56
chaîne
  de notification, 239
change_bit(), 132
change_nexthops(), 345
checksum, 36
checksum.h
  l.54–91, 320
  l.6–18, 422
  l.93–135, 432
checksum.h/net
  l.60–64, 427
  l.71–77, 427
checksum.S
  l.6–252, 422
CHECKSUM_HW, 155
CHECKSUM_NONE, 155
CHECKSUM_UNNECESSARY, 155
CIDR, 107
circuit, 4
circuit switching, 4
clé, 343
class_device (struct), 222
class_device_create_file(), 252
class_device_register(), 252
class_device_unregister(), 252
close(), 69
CMSG_ALIGN(), 90
CMSG_DATA(), 90
CMSG_FIRSTHDR(), 90
CMSG_LEN(), 90
CMSG_NXTHDR(), 90
CMSG_SPACE(), 90
cmsghdr (struct), 89, 520
codage
  binaire simple, 24
  Manchester, 24
collision, 42
commande
  ifconfig, 97, 206, 215, 218
  ip, 324, 327
  route, 324
communication
  connectée, 10, 66
  non connectée, 66
commutation

automatique, 4
de circuits, 4
de message, 5
manuelle, 4
par paquets, 5
compare_keys(), 375
compiler.h
l.54–61, 166
COMPLETE, 633
concentrateur, 30
CONFIG_INET, 327
CONFIG_IP_ADVANCED_ROUTER, 327
CONFIG_IP_MULTIPLE_TABLES, 327
CONFIG_IP_ROUTE_VERBOSE, 393
CONFIG_NETLINK, 327
CONFIG_RTNETLINK, 327
confirmation, 39
connect(), 74
connecteur
BNC, 28, 46
connection_based(), 533
connexion, 62
à distance, 60
distribuée, 60
container_of(), 477
contrôle
de flux
avec retour d'information, 41
basé sur le débit, 41
contrôle des flux, 35
cookie, 525
copy_from_user(), 126
copy_skb_header(), 187
copy_to_user(), 126
cordon
AUI, 47
core/, 118
couche, 7
(sous-) 2a, 48
(sous-) 2b, 48
(sous-) LLC, 48
(sous-) MAC, 48
(sous-) de contrôle d'accès au canal, 48
d'application, 10
de liaison, 9
de présentation, 10
de réseau, 9
de session, 10
de transport, 9
inférieure, 9
Internet, 11
physique, 9
supérieure, 9
courrier électronique, 61
Cox, Alan, 145
CRC, 101
credentials, 87
Crocker, 57
crypto/, 118
CSMA, 43
CSNET, 30
CSP, 43
csum_add(), 427
csum_block_add(), 427
csum_fold(), 432
csum_partial(), 422
csum_partial_copy_fromiovecend(), 564
csum_tcpudp_magic(), 432
csum_tcpudp_nofold(), 432
CTSS, 13
current_text_addr(), 168

débit, 23
délai, 40
démon
réseau, 74
démultiplexage, 10
DARPA, 16
data
rate, 23
data link layer, 9
datagram.c
l.116–195, 529
l.197–200, 537
l.215–297, 533
l.390–436, 535
l.58–64, 533
l.66–114, 531
datagram_poll(), 480
datagramme, 58
DATAPORT, 269
Davies, Donald Watts, 14
daytime, 74
dcache.h
l.111–118, 466
l.27–38, 473
l.83–109, 466
DECLARE_MUTEX(), 138
decnet/, 119
default_rebuild_header(), 244

DEFINE_SNMP_STAT(), 380
DEFINE_WAIT(), 532
del_timer(), 128
deliver_skb(), 317
dentry, 461
dentry (struct), 466
dentry_operations (struct), 466
descripteur
    de couche transport, 403
    de fichier, 461
    de nœud d'information, 460
    de protocole, 398
    de socket, 468
    de tampon de socket, 147
dev.c
    l.1004–1015, 415
    l.1017–1025, 602
    l.1027–1070, 602
    l.1072–1106, 594
    l.1108–1129, 593
    l.1133–1189, 592
    l.1218–1333, 590
    l.129–159, 305
    l.1340, 301
    l.1341, 302
    l.1351, 599
    l.1405–1481, 302
    l.1498–1506, 316
    l.1554–1559, 317
    l.1625–1714, 314
    l.166–184, 234
    l.1716–1762, 312
    l.1764-1805, 311
    l.185, 200
    l.187, 234
    l.192–200, 238
    l.207–211, 239
    l.2151–2161, 219
    l.2176–2188, 218
    l.236–240, 599
    l.242–284, 307
    l.2656–2672, 246
    l.2674, 243
    l.2676–2677, 250
    l.2678–2685, 245
    l.2687–2816, 241
    l.2730–2783, 619
    l.2887–2892, 453
    l.2892, 250
    l.2893–2958, 249
    l.290–344, 308
    l.2960–2985, 274
    l.2994–3079, 615
    l.3153–3155, 307
    l.3163–3181, 301
    l.3190–3191, 310
    l.346–353, 260
    l.382–407, 273
    l.410–439, 258
    l.476–498, 237
    l.523–545, 247
    l.640–653, 244
    l.655–718, 236
    l.799–805, 244
    l.808–873, 281
    l.875–934, 612
    l.900–918, 232
dev_3c501, 232
dev_activate(), 283
dev_add_pack(), 307
dev_alloc_name(), 236
dev_alloc_skb(), 197
dev_base, 200
dev_base_lock, 234
dev_boot_phase, 243
dev_boot_setup[], 260
dev_close(), 612
dev_deactivate(), 614
dev_get_index(), 208
dev_hold(), 245
dev_index_head[], 238
dev_init_scheduler(), 247
dev_kfree_skb(), 166
dev_mc_discard(), 618
dev_mc_list (struct), 218
dev_mcast.c
    l.202–220, 618
dev_name_hash(), 238
dev_name_head[], 238
dev_new_index(), 246
dev_open(), 281
dev_put(), 618
dev_queue_xmit(), 590
dev_queue_xmit_nit(), 602
dev_remove_pack(), 308
dev_set_allmulti(), 218
dev_set_promiscuity(), 219
dev_shutdown(), 617
dev_valid_name(), 244
dev_watchdog(), 609

dev_watchdog_down(), 614
dev_watchdog_init(), 249
dev_watchdog_up(), 285
device.h
    l.28, 252
device_initcall(), 230
devinet.c
    l.1482–1492, 371
devinet_init(), 371
devprobe2 (struct), 261
diffusion
    générale, 100
    individuelle, 27
    multidestinataire, 100
    multidiffusion, 100
    restreinte, 100
divert.h
    l.111–130, 317
    l.14–34, 222
divert_blk (struct), 222
divert_frame(), 317
DIVERT_PROTO_ICMP, 222
DIVERT_PROTO_IP, 222
DIVERT_PROTO_NONE, 222
DIVERT_PROTO_TCP, 222
DIVERT_PROTO_UDP, 222
DNS, 61
do_softirq(), 131
Documentation/, 118
donnée
    ancillaire, 87
    auxiliaire, 87
    de contrôle, 87
données
    de niveau n, 7
dorsale, 25, 29, 30
down(), 138
drivers/, 118
drop, 30
DRV_NAME, 277
dst.c
    l.115–138, 394
dst.h
    l.1–9, 364
    l.137–143, 183
    l.145–152, 165
    l.188–192, 554
    l.219–232, 585
    l.234–248, 397
    l.29–80, 364
    l.83–101, 366
dst_alloc(), 394
dst_clone(), 183
dst_confirm(), 554
dst_entry (struct), 364
DST_HOST, 365
dst_input(), 397
DST_NOHASH, 365
DST_NOPOLICY, 365
DST_NOXFRM, 365
dst_ops (struct), 366
dst_output(), 585
dst_release(), 165
DT_SOCK, 466
durée
    de vie, 103
dv.c
    l.48–76, 245
    l.6–12, 245
    l.78–94, 618

EACCES, 68, 513, 553
EADDRINUSE, 513
EADDRNOTAVAIL, 513
EAFNOSUPPORT, 71, 489
EAGAIN, 277, 528, 531, 545, 563
EBADF, 76, 84, 502, 507
ECMA, 3
econet/, 119
EDSTADDRREQ, 552
EEXIST, 244, 344, 346, 451
EFAULT, 186, 535, 536, 561
EINVAL, 68, 71, 237, 244, 343, 351, 362, 392, 394, 396, 455, 488, 503, 513, 536, 552, 554, 579, 588
EIO, 244
EIOCBQUEUED, 523, 543
el1_close(), 614
EL1_DATAPORT, 269
EL1_DATAPTR, 269
el1_get_stats(), 318
EL1_IO_EXTENT, 273
el1_probe(), 271
el1_probe1(), 274
EL1_RXPTR, 269
EL1_SAPROM, 269
el_debug, 277
el_interrupt(), 292
el_open(), 286
el_receive(), 296

el_reset(), 287
el_start_xmit(), 604
el_timeout(), 610
EMFILE, 68
EMSGSIZE, 551, 560, 628
en-queue
  de trame, 36
en-tête
  de message, 88, 520
  de protocole, 7, 12
  de trame, 36
  matériel, 199
  physique, 199
  pseudo, 112
endfor_nexthops(), 345
ENETUNREACH, 362
ENFILE, 68, 71, 237, 489, 495
Engelbart, Doug, 58
ENOBUFS, 343, 346, 354, 394, 451, 498, 560, 563
ENODEV, 208, 260, 273, 277, 283, 481, 617
ENOENT, 346
ENOMEM, 68, 237, 246, 272, 371, 473, 495, 587, 592, 594, 631
ENOTCOMM, 533
ENOTCONN, 84, 503
ENOTSOCK, 76, 84, 502
ENXIO, 273, 481
EOPNOTSUPP, 76
EPERM, 498
EPIPE, 563
EPROTONOSUPPORT, 68
EPROTONOTSUPPORT, 498
err.h
  l.21–29, 233
  l.8–19, 272
ERR_PTR(), 272
ESOCKNOTSUPPORT, 498
ESPIPE, 479
eth.c
  l.153–209, 298
  l.266–288, 261
  l.291–307, 262
ETH_ALEN, 256
ETH_DATA_LEN, 256
ETH_FRAME_LEN, 256
eth_hdr(), 299
ETH_HLEN, 256
ETH_P_ALL, 307
ETH_P_ARP, 257
ETH_P_IP, 257
ETH_P_LOOP, 257
eth_type_trans(), 298
ETH_ZLEN, 256
ether_setup(), 261
Ethernet, 32, 59
  épais, 46
  fin, 46
  rapide, 28
ethernet/, 119
ethhdr (struct), 257
ethif_probe2(), 258
ethtool.h
  l.16–30, 266
  l.264–354, 264
ethtool_cmd (struct), 266
ethtool_ops (struct), 264

fanion
  de signalisation, 38
Fast Ethernet, 28
fasync_struct (struct), 468
fc_setup(), 261
FDDI, 261
fddi_setup(), 261
fddi_type_trans(), 298
FDM, 42
fdopen(), 77
feed-back based flow control, 41
fenêtre glissante, 40
FIB, 327
fib_alias, 343
fib_create_info(), 348
fib_find_alias(), 344
fib_find_node(), 343
fib_frontend.c
  l.127–151, 513
  l.153–230, 395
  l.50–55, 332
  l.527–586, 333
  l.588–607, 333
  l.99–125, 579
fib_hash.c
  l.143–188, 354
  l.201–241, 347
  l.243–273, 359
  l.384–390, 347
  l.392–405, 343
  l.407–425, 344
  l.427–576, 340

l.48–49, 335
l.51–55, 330
l.57–73, 330
l.579–656, 355
l.75–78, 329
l.799–824, 347
l.826–860, 335
l.90–93, 343
l.95, 346
l.97, 343
fib_hash_alloc(), 352
fib_hash_free(), 352
fib_hash_init(), 335
fib_hash_lock, 346
fib_hash_move(), 352
fib_hash_size, 352
fib_inetaddr_event(), 333
fib_inetaddr_notifier, 333
fib_info (struct), 331
fib_info_cnt, 352
fib_info_lock, 344
fib_info_put(), 345
fib_insert_node(), 347
fib_lookup(), 362
fib_lookup.c
l.8–15, 343
fib_netdev_event(), 333
fib_netdev_notifier, 333
fib_nh (struct), 330
fib_node (struct), 330
fib_props[], 351
fib_release_info(), 344
fib_res_put(), 514
fib_result, 358
fib_rules.c, 327
fib_select_default(), 580
fib_semantic_match(), 360
fib_semantics.c
l.141–156, 345
l.158–174, 344
l.50, 344
l.503–570, 352
l.53–54, 352
l.572–763, 348
l.72–82, 345
l.765–832, 360
l.85–138, 351
fib_table (struct), 328
fib_validate_source(), 395
fichier
descripteur, 461
numéro, 461
file (struct), 465
file.h
l.19–34, 506
file_operations (struct), 464
file_system_type (struct), 467
files_struct (struct), 506
filesystems.c
l.53–84, 467
filp_close(), 507
filter.h
l.21–32, 410
l.40–46, 410
l.48–51, 415
FIRST_IN, 633
flowi (struct), 357
fn_alias_kmem, 335
fn_hash (struct), 329
fn_hash_delete(), 355
fn_hash_insert(), 340
fn_hash_kmem, 335
fn_hash_lookup(), 359
fn_new_zone(), 347
fn_rehash_zone(), 354
fn_zone (struct), 330
forwarding, 105
frag_alloc_queue(), 637
FRAG_CB(), 641
frag_free_queue(), 643
frag_kfree_skb(), 643
fragment, 104
de tampon
liste, 149
tableau, 149
fragment.c
l.644–679, 634
fragmentation, 9, 104
des données, 102
frame, 36, 58
Frame Relay, 59
free_divert_blk(), 618
free_fib_info(), 345
free_irq(), 130
free_netdev(), 274
fs.h, 468
l.1137–1146, 467
l.429–489, 463
l.577–603, 465
l.754–800, 461

l.844–858, 466
l.903–934, 464
l.969–995, 462
fs/, 118
FTP, 56
full_name_hash(), 238
FZ_HASHMASK(), 330
fz_key(), 343
FZ_MASK(), 330
FZ_MASK_DIVISOR, 343

gateway, 26
gen_stats.h
l.16–53, 212
generic.h
l.1–172, 224
generic_shutdown_super(), 474
get_free_pages(), 125
get_sb_pseudo(), 471
get_stats()/loopback.c, 231
getsockopt(), 91
GFP_ATOMIC, 125
GFP_DMA, 125
GFP_KERNEL, 125
GFP_USER, 125
gnet_stats_basic (struct), 212
gnet_stats_queue (struct), 212
gnet_stats_rate_est (struct), 212
GP_HIGH, 269
GP_LOW, 269

handle_bridge(), 317
handle_diverter(), 317
header, 36
headroom, 148
hh_cache (struct), 222
HH_DATA_ALIGN(), 222
HH_DATA_MOD, 222
HH_DATA_OFF(), 222
HIPPI, 156
hippi_setup(), 261
hlist_entry(), 238
hlist_for_each(), 238
hlist_node (struct), 214
hook, 381
hop, 103
htonl(), 73
htons(), 73
HTTP, 61
hub, 30
HW-IRQ, 129

IAB, 58
IANA, 62, 110
ICCB, 58
ICMP, 59, 103
numéro de protocole IP, 110
ICMP_EXC_FRAGTIME, 643
icmp_protocol, 449
icmp_send(), 402
ICMP_TIME_EXCEEDED, 643
identifiant
de système terminal, 106
IEEE, 48
IETF, 58
if.h
l.26, 206
l.29–50, 214
l.52–81, 215
l.84–103, 259
if_arp.h, 154
l.28–89, 216
if_ether.h
l.108–111, 299
l.35–93, 256
l.6–33, 256
l.95–104, 257
IF_GET_IFACE, 215
IF_GET_PROTO, 215
IF_IFACE_E1, 215
IF_IFACE_SYNC_SERIAL, 215
IF_IFACE_T1, 215
IF_IFACE_V24, 215
IF_IFACE_V35, 215
IF_IFACE_X21, 215
if_packet.h
l.22–31, 155
IF_PORT_100BASEFX, 206
IF_PORT_100BASET, 206
IF_PORT_100BASETX, 206
IF_PORT_10BASE2, 206
IF_PORT_10BASET, 206
IF_PORT_AUI, 206
IF_PORT_UNKNOWN, 206
IF_PROTO_CISCO, 215
IF_PROTO_FR, 215
IF_PROTO_FR_ADD_ETH_PVC, 215
IF_PROTO_FR_ADD_PVC, 215
IF_PROTO_FR_DEL_ETH_PVC, 215
IF_PROTO_FR_DEL_PVC, 215

IF_PROTO_FR_ETH_PVC, 215
IF_PROTO_FR_PVC, 215
IF_PROTO_HDLC, 215
IF_PROTO_HDLC_ETH, 215
IF_PROTO_PPP, 215
IF_PROTO_X25, 215
ifconfig, 97, 279
IFF_802_1Q_VLAN, 215
IFF_ALLMULTI, 215
IFF_AUTOMEDIA, 215
IFF_BROADCAST, 215
IFF_DEBUG, 215
IFF_DYNAMIC, 215
IFF_EBRIDGE, 215
IFF_LOOPBACK, 215
IFF_MASTER, 215
IFF_MULTICAST, 215
IFF_NOARP, 215
IFF_NOTRAILERS, 215
IFF_POINTOPOINT, 215
IFF_PORTSEL, 215
IFF_PROMISC, 215
IFF_RUNNING, 215
IFF_SLAVE, 215
IFF_UP, 214
IFF_VOLATILE, 214
ifmap (struct), 259
IFNAMESIZ, 206
igmhdr (struct), 153
IGMP, 60
igmp_protocol, 449
illegal_highdma(), 593
IMP, 15
in-band, 78
in.h, 72, 73
    l.24–49, 445
    l.244–249, 393
    l.246, 386
IN_DEV_FORWARD(), 393
in_dev_get(), 392
in_dev_put(), 395
in_device (struct), 387
in_ifaddr (struct), 387
in_route.h
    l.30, 552
INADDR_ANY, 73
INADDR_LOOPBACK, 73
include/, 118
index
    de périphérique réseau, 200
inet_add_protocol(), 398
inet_addr_type(), 513
inet_autobind(), 545
inet_bind(), 511
inet_create(), 495
inet_dgram_ops, 480
inet_family_ops, 443
inet_init(), 399, 444, 453
inet_opt (struct), 433
inet_protos[], 398
inet_protosw (struct), 445
INET_PROTOSW_PERMANENT, 447
INET_PROTOSW_REUSE, 447
inet_register_protosw(), 452
inet_release(), 484
inet_reset_saddr(), 505
inet_sendmsg(), 544
inet_shutdown(), 502
inet_sk(), 435
inet_sock (struct), 435
inetd, 74
inetdevice.h
    l.143–154, 392
    l.37–59, 387
    l.85–99, 387
inetsw[], 456
inetsw_array[], 448
INETSW_ARRAY_LEN, 456
inetsw_lock, 453
infirmation, 39
init.h
    l.148–156, 140
    l.7–40, 139
    l.78–87, 140
    l.94, 140
    l.97, 140
init/, 118
init_ipv4_mibs(), 456
INIT_LIST_HEAD(), 248
init_module(), 232
init_sync_kiocb(), 523
init_timer(), 128
inode, 460
inode (struct), 463
int16_t, 123
int32_t, 123
int64_t, 123
int8_t, 123
interface, 7
    Netlink, 327

Internet, 26, 33
  layer, 11
internet, 26
Internet Society, 58
interréseau, 26
interrupt.h
  l.108, 310
  l.16–34, 294
  l.79–93, 131
interruption
  matérielle, 129
    lente, 130
    rapide, 130
ioctl(), 215, 221
iovec (struct), 86, 520
iovec.c
  l.147–256, 564
  l.77–99, 535
IP, 59
ip
  commande, 327
ip.h
  l.110–145, 433
  l.147, 583
  l.151–163, 435
  l.155, 379
  l.179–185, 584
  l.236–247, 505
  l.314–321, 498
  l.41, 573
  l.57, 630
  l.75, 632
  l.78–81, 499
  l.88–106, 581
ip.h/linux
  l.167–187, 319
  l.21–37, 320
  l.6–19, 319
ip.h/net
  l.69–73, 320
ip_append_data(), 555
IP_CE, 320
ip_cmsg_recv(), 529
ip_cmsg_send(), 552
ip_copy_metadata(), 629
ip_defrag(), 634
ip_dev_find(), 579
IP_DF, 320
ip_dont_fragment(), 584
ip_evictor(), 635
ip_expire()(), 644
ip_fast_csum(), 320
ip_fib.h, 327
  l.122–148, 328
  l.158–164, 362
  l.166–170, 580
  l.23–40, 336
  l.254–258, 345
  l.260–268, 514
  l.44–59, 330
  l.61–86, 331
  l.93–102, 358
ip_fib_init(), 333
ip_fib_local_table, 332
ip_fib_main_table, 332
ip_find(), 636
ip_finish_output(), 586
ip_finish_output2(), 586
ip_forward(), 397
ip_frag_create(), 637
ip_frag_destroy(), 643
ip_frag_intern(), 637
ip_frag_mem, 634
ip_frag_nqueues, 644
ip_frag_queue(), 638
ip_frag_reasm(), 641
IP_FRAG_TIME, 632
ip_fragment(), 624
ip_fragment.c
  l.103, 636
  l.105–121, 644
  l.123–127, 636
  l.169, 634
  l.171–178, 643
  l.180–186, 643
  l.188–196, 637
  l.199–220, 643
  l.222–226, 642
  l.228–241, 644
  l.243–280, 635
  l.282–310, 644
  l.312–352, 637
  l.354–384, 637
  l.386–412, 636
  l.414–551, 638
  l.46–57, 634
  l.554–642, 641
  l.64–70, 641
  l.681–690, 645
  l.72–94, 633

l.96–101, 633
ip_generic_getfrag(), 564
IP_INC_STATS_BH(), 379
ip_init(), 456
ip_input.c
l.149–153, 379
l.199–264, 399
l.266–283, 397
l.285–355, 382
l.357–429, 378
ip_local_deliver(), 397
ip_local_deliver_finish(), 399
ip_local_error(), 560
IP_MF, 320
ip_mr_input(), 397
IP_OFFSET, 320
ip_options (struct), 581
ip_options.c
l.202–238, 630
ip_options_fragment(), 630
ip_output(), 586
ip_output.c
l.1107–1213, 581
l.118–125, 584
l.1323–1330, 305
l.1336–1346, 456
l.172–214, 586
l.216–225, 586
l.287–295, 586
l.386–418, 629
l.420–667, 624
l.669–684, 564
l.697–971, 555
l.95–100, 585
ip_packet_type, 305
IP_PMTUDISC_DO, 499
IP_PMTUDISC_DONT, 499
IP_PMTUDISC_WANT, 499
ip_push_pending_frames(), 581
ip_rcv(), 378
ip_rcv_finish(), 382
ip_recv_error(), 528
ip_route_input(), 384
ip_route_input_slow(), 388
ip_route_output_flow(), 572
ip_route_output_slow(), 573
ip_rt_bug(), 394
ip_rt_init(), 368
ip_rt_max_size, 371
ip_rt_put(), 554
ip_select_ttl(), 584
ip_send_check(), 585
ip_statistics, 379
ipc/, 118
IPCB(), 630
ipcm_cookie (struct), 551
IPCORK_OPT, 583
ipfrag_hash_rnd, 636
ipfrag_init(), 645
ipfrag_skb_cb, 641
iphdr (struct), 319
IPng, 60
IPPROTO_AH, 445
IPPROTO_COMP, 445
IPPROTO_EGP, 445
IPPROTO_ESP, 445
IPPROTO_GRE, 445
IPPROTO_ICMP, 445
IPPROTO_IDP, 445
IPPROTO_IGMP, 445
IPPROTO_IP, 445
IPPROTO_IPIP, 445
IPPROTO_IPV6, 445
IPPROTO_MAX, 445
IPPROTO_PIM, 445
IPPROTO_PUP, 445
IPPROTO_RAW, 445
IPPROTO_RSVP, 445
IPPROTO_SCTP, 445
IPPROTO_TCP, 71, 445
IPPROTO_UDP, 71, 445
ipq (struct), 633
ipq_hash[], 633
IPQ_HASHZ, 633
ipq_kill(), 644
ipq_put(), 642
ipq_unlink(), 644
ipqhashfn(), 636
IPsec, 60
ipstats_mib (struct), 380
IPSTATS_MIB_INDISCARDS, 380
IPSTATS_MIB_INRECEIVES, 380
IPTOS_LOWDELAY, 320
IPTOS_RT_MASK, 386
IPTOS_THROUGHPUT, 320
IPTOS_TOS(), 320
IPTOS_TOS_MASK, 320
IPv4, 60
ipv4/, 118
ipv4_config, 498

ipv4_config (struct), 498
ipv4_dst_ops, 366
ipv4_proc_init(), 457
IPv6, 60
ipv6.h
    l.292–298, 435
ipv6/, 119
ipv6_only_sock(), 435
ipv6hdr (struct), 153
ipx.h, 73
ipx/, 119
irda/, 119
IRQ_HANDLED, 294
IRQ_NONE, 294
IRQ_RETVAL(), 294
irqreturn_t, 294
IRTF, 58
IS_ERR(), 233
isa_probes[], 261
ISO, 9

Kahn, 56
kern_rta (struct), 336
kernel.h
    l.232–242, 477
kernel/, 118
key/, 119
kfree(), 126
kfree_skb(), 164
kfree_skbmem(), 162
kill_anon_super(), 474
kiocb (struct), 522
kiocb_to_siocb(), 525
Kleinrock, Leonard, 14
kmalloc(), 125
kmap_skb_frag(), 186
kmem_cache_alloc(), 127
kmem_cache_create(), 126
kmem_cache_free(), 127
kmem_cache_t, 127
kstatfs (struct), 475
kunmap_skb_frag(), 186
Kuznetsov, Alexey, 327

LAN, 25
lapb/, 119
LAST_IN, 633
lib/, 118
libfs.c
    l.192–237, 471
    l.21–27, 475
Licklider, Joseph Carl Robnett, 15
likely(), 166
list.h
    l.18–30, 210
    l.298–306, 250
    l.314–321, 498
    l.37–39, 248
    l.427–438, 498
    l.503–505, 214
    l.631–635, 238
    l.70–81, 310
list_add_tail(), 310
list_entry(), 498
list_for_each_rcu(), 498
list_head, 210
list_splice_init(), 250
listen(), 76
little_endian.h
    l.1–9, 227
LL_MAX_HEADER, 223
LL_RESERVED_SPACE(), 560
llc/, 119
llc_rcv(), 299
lock, 134
lock_sock(), 418
loff_t, 125
logiciel, 2
loopback, 73, 230
LOOPBACK(), 393
loopback.c
    l.161–185, 231
    l.198–215, 230
    l.217–231, 231
loopback_dev, 230
loopback_init(), 231
lpr, 72
lsocket.c
    l.276–293, 476
    l.295–299, 477
ltalk_setup(), 261
LXR, 121
lxr, vii

métrique, 281
MAN, 25
man
    byteorder, 73
    cmsg, 90
    ddp(7), 67

- ifconfig, 280
- ip(4), 73
- ip(7), 67
- netlink(7), 67
- packet(7), 67
- recv, 80
- sendmsg, 79
- unix(7), 67
- x25(7), 67

manage.c
- l.218–232, 130
- l.279–310, 129

Marill, Tom, 2, 13
marque, 23
marqueur
- de début, 22
- de fin, 22

masque
- réseau, 108

matériel, 2
MAX_ADDR_LEN, 217
MAX_DIVERT_PORTS, 222
MAX_INET_PROTOS, 398
MAX_SCHEDULE_TIMEOUT, 418
MAX_SKB_FRAGS, 149
media, 9
medium, 2, 21
memcpy_fromiovecend(), 564
memcpy_toiovec(), 535
message, 58
message switching, 5
Metcalfe, Bob, 32
metric, 281
minuteur
- de réassemblage, 104

mm/, 118
modèle
- en couches, 7
- OSI, 8, 9
- réseau, 7

mode
- de promiscuité, 142

modpost.h
- l.71–81, 139

module, 138
module (struct), 139
module.h
- l.302–315, 489
- l.317–327, 489

module_init(), 140
module_put(), 489
mot, 108
move_addr_to_kernel(), 510
move_addr_to_user(), 523
MSG_CONFIRM, 79
MSG_CTRUNC, 88
MSG_DONTROUTE, 79
MSG_DONTWAIT, 79
MSG_EOR, 88
MSG_ERRQUEUE, 80, 88
MSG_NOSIGNAL, 79, 80
MSG_OOB, 79, 88
MSG_PEEK, 79
MSG_TRUNC, 80, 88
MSG_WAITALL, 80
msghdr (struct), 88, 520
MTU, 216
multicast, 60
MULTICAST(), 386
MULTICS, 13
multihoming, 108
multiplexage, 10
mutex, 138

NAK, 39
NCP, 13
neigh_confirm(), 554
neigh_ops (struct), 588
neighbour.h
- l.316–236, 554

Net-1, 144
Net-2, 145
Net-3, 145
Net-4, 145
net-sysfs.c
- l.188–201, 252
- l.258–283, 253
- l.286–289, 253
- l.386–399, 619
- l.389–395, 252

net-tools, 279
net.h
- l.124–160, 469
- l.162–171, 443
- l.29, 443
- l.49–55, 469
- l.61–63, 469
- l.65–91, 444
- l.93–115, 468

net/, 118

Net/one, 33
NET_BH, 131
net_class, 252
net_class_attributes[], 252
net_dev_init(), 301, 307, 310
net_device (struct), 201
net_device_stats (struct), 209
net_disable_timestamp(), 415
net_families[], 444
net_family_lock, 451
net_family_lockct, 451
net_family_read_lock(), 489
net_family_read_unlock(), 489
net_family_write_lock(), 451
net_family_write_unlock(), 451
net_init.c, 261
  l.108–142, 233
  l.144–149, 615
  l.55–70, 200
  l.73–105, 263
net_local (struct), 210, 277
net_olddevs_init(), 230
net_proto_family (struct), 443
net_protocol (struct), 398
net_rx_action(), 311
NET_RX_BAD, 300
NET_RX_CN_HIGH, 300
NET_RX_CN_LOW, 300
NET_RX_CN_MOD, 300
NET_RX_DROP, 300
NET_RX_SOFTIRQ, 131
NET_RX_SUCCESS, 300
net_set_todo(), 245
net_timestamp(), 602
net_todo_list, 245
net_todo_list_lock, 250
net_todo_run_mutex, 250
NET_TX, 208
NET_TX_SOFTIRQ, 131
NET_XMIT_BYPASS, 594
NET_XMIT_SUCCESS, 594
NetBIOS, 55
NETDEV_ALIGN_CONST, 263
netdev_boot_base(), 258
netdev_boot_setup (struct), 259
netdev_boot_setup_add(), 260
netdev_boot_setup_check(), 273
NETDEV_BOOT_SETUP_MAX, 259
netdev_chain, 239
NETDEV_CHANGE, 240
NETDEV_CHANGEADDR, 240
NETDEV_CHANGEMTU, 240
NETDEV_CHANGENAME, 240
NETDEV_DOWN, 240
netdev_ethtool_ops, 278
netdev_get_drvinfo(), 278
netdev_get_msglevel(), 278
NETDEV_HASHBITS, 238
netdev_max_backlog, 301
netdev_nit, 599
netdev_priv(), 264
NETDEV_REGISTER, 240
netdev_register_sysfs(), 251
netdev_run_todo(), 249
netdev_rx_stat, 599
netdev_set_msglevel(), 278
netdev_state_t, 207
NETDEV_TX_BUSY, 599
NETDEV_TX_LOCKED, 599
NETDEV_TX_OK, 599
NETDEV_UNREGISTER, 240
netdev_unregister_sysfs(), 619
NETDEV_UP, 240
netdev_wait_allrefs(), 619
netdevice.h
  l.103–139, 209
  l.142–151, 206
  l.156–160, 396
  l.162–174, 599
  l.164–168, 395
  l.179–190, 218
  l.192–211, 222
  l.213–224, 560
  l.226–240, 207
  l.236–240, 599
  l.243–251, 259
  l.254–491, 201
  l.493–494, 263
  l.496–501, 264
  l.503–506, 273
  l.509–516, 304
  l.54–60, 594
  l.563–579, 300
  l.581, 301
  l.585–604, 601
  l.6–26, 201
  l.606–609, 287
  l.61, 393
  l.611–619, 609
  l.62–68, 300

l.621–628, 606
l.630–633, 596
l.635–638, 309
l.690–693, 618
l.695, 614
l.696, 245
l.726–730, 283
l.74, 217
l.799–805, 309
l.807–824, 309
l.81–94, 223
l.826–832, 309
l.879–882, 313
Netfilter, 381
netfilter.h
l.121–157, 381
netif_device_present(), 283
NETIF_F_HW_CSUM, 220
NETIF_F_IP_CSUM, 220
NETIF_F_NO_CSUM, 220
netif_poll_enable(), 313
netif_queue_stopped(), 207, 596
netif_receive_skb(), 314
netif_running(), 207, 309
netif_rx(), 302
netif_rx_schedule(), 309
netif_rx_schedule_prep(), 309
netif_rx_stats, 599
netif_schedule(), 601
netif_schedule(dev), 599
netif_start_queue(), 208, 287
netif_stop_queue(), 207, 606
netif_wake_queue(), 208, 609
Netlink, 327
netlink, 234
netlink.c
l.54–59, 234
l.61–66, 235
netlink.h
l.800, 235
l.803–806, 235
netlink/, 118
NETLINK_ROUTE, 327
netlink_skb_parms (struct), 340
NETREG_REGISTERED, 214
NETREG_REGISTERING, 214
NETREG_RELEASED, 214
NETREG_UNINITIALIZED, 214
NETREG_UNREGISTERED, 214
NETREG_UNREGISTERING, 214
netrom/, 119
netstamp_needed, 415
netstat_attrs, 253
netstat_group, 253
network, 2
layer, 9
nf_debug_ip_local_deliver(), 400
NF_HOOK(), 381
NF_IP_LOCAL_IN, 397
NF_IP_POST_ROUTING, 586
nf_reset(), 400
NIC, 58, 61
niveau
administrateur, v
conception du noyau réseau, v
programmation réseau, v
utilisateur, v
NLM_F_ACK, 339
NLM_F_APPEND, 339
NLM_F_ATOMIC, 339
NLM_F_CREATE, 339
NLM_F_DUMP, 339
NLM_F_ECHO, 339
NLM_F_EXCL, 339
NLM_F_MATCH, 339
NLM_F_MULTI, 339
NLM_F_REPLACE, 339
NLM_F_REQUEST, 339
NLM_F_ROOT, 339
nlmsghdr (struct), 339
NLS, 58
no_llseek(), 479
non-acknowledge, 39
noop_dequeue(), 600
noop_enqueue(), 595
noop_qdisc, 248
noop_qdisc_ops, 248
noop_requeue(), 600
noqueue_qdisc, 285
noqueue_qdisc_ops, 285
norme
IEEE 802.3, 45
notifier-chain, 239
notifier.h
l.1–8, 239
l.14–19, 239
l.28–35, 241
l.45–58, 240
notifier_block (struct), 239
notifier_call_chain(), 240

NOTIFY_DONE, 241
NOTIFY_STOP_MASK, 241
Novell, 33
NPROTO, 443
ntohl(), 73
ntohs(), 73
numéro
  de fichier, 461
  de socket, 67
NWG, 57
nœud d'information, 460

offsetof(), 568
opération
  atomique, 132
open.c
  l.1018–1043, 506
  l.986–1014, 507
option
  IP, 103
ordre
  des octets
    grand boutien, 73
    petit boutien, 73
    réseau, 73
OSI, 8
OUI, 100
out-of-band, 78

périphérique
  activation, 279
  désactivation, 279
  en boucle, 230
  enregistrement, 279
  fermeture, 279
  ouverture, 279
  réseau
    en boucle, 200
  retrait, 279
packet, 36
packet switching, 5
packet/, 118
PACKET_BROADCAST, 155
PACKET_FASTROUTE, 155
PACKET_HOST, 155
PACKET_LOOPBACK, 155
PACKET_MULTICAST, 155
PACKET_OTHERHOST, 155
packet_type (struct), 304
PAN, 25
paquet, 14, 36
  Internet, 108
  IP, 108
partie
  basse, 131
  haute, 130
passerelle, 26, 103
PCI, 12
PDU, 12
peer, 62
peer-to-peer, 28
PF_APPLETALK, 67, 442
PF_ASH, 442
PF_ATMPVC, 67, 442
PF_ATMSVC, 442
PF_AX25, 67, 442
PF_BLUETOOTH, 442
PF_BRIDGE, 442
PF_DECnet, 442
PF_ECONET, 442
PF_INET, 67, 442
PF_INET6, 67, 442
PF_IPX, 67, 442
PF_IRDA, 442
PF_KEY, 442
PF_LLC, 442
PF_LOCAL, 67, 442
PF_MAX, 442
PF_NETBEUI, 442
PF_NETLINK, 67, 327, 442
PF_NETROM, 442
PF_PACKET, 67, 442
PF_PPPOX, 442
PF_ROSE, 442
PF_ROUTE, 442
PF_SECURITY, 442
PF_SNA, 442
PF_UNIX, 67, 442
PF_UNSPEC, 442
PF_WANPIPE, 442
PF_X25, 67, 442
pfifo_fast_dequeue(), 600
pfifo_fast_enqueue(), 595
pfifo_fast_ops, 284
pfifo_fast_requeue(), 601
physical layer, 9
pile
  réseau, 8
ping, 59
pkt_sched.h

l.16–17, 285
l.231–235, 596
l.4–24, 595
plaque
de cache, 126
point
de synchronisation, 10
point d'ancrage, 381
polling, 80
port, 30, 62
Posix.1g, 90
posix_types.h
l.1–39, 122
posix_types.h/linux, 121
PPP, 59
préfixe
réseau, 107
prepare_to_wait_exclusive(), 532
presentation layer, 10
prise
RJ45, 46
probe_list2(), 260
proc/, 457
process_backlog(), 312
processor.h
l.36–40, 168
processus, 129
PROT_SOCK, 513
proto (struct), 447
proto_ops (struct), 469
protocol
handler, 398
protocol.c
l.51, 398
l.54–74, 398
protocol.h
l.33, 398
l.36–41, 398
l.59–76, 445
l.77–78, 447
protocole, 2, 13
à détection de porteuse, 43
à diffusion
restreinte, 60
unique, 60
1822, 16
en-tête, 7
suffixe, 7
pskb_copy(), 188
pskb_expand_head(), 193
pskb_may_pull(), 190
pskb_pull(), 190
pskb_trim(), 195
pt_regs (struct), 293
PTR_ERR(), 233
ptrace.h
l.23–42, 293
ptype_all, 305
ptype_base[], 305
ptype_lock, 305

Qdisc (struct), 211
QDISC_ALIGN_CONST, 285
Qdisc_class_ops (struct), 213
qdisc_create_dflt(), 284
qdisc_lock_tree(), 247
Qdisc_ops (struct), 213
qdisc_reset(), 614
qdisc_restart(), 596
qdisc_run(), 596
qdisc_tree_lock, 247
qdisc_unlock_tree(), 248
qstr (struct), 473

réassemblage, 104
régulation des flux, 35
réseau
étendu, 25
à diffusion, 26
démon, 74
distribué, 6
en étoile, 30
en anneau, 29
en bus, 29
familial, 25
hiérarchisé, 6
local, 25
longue distance, 25
métropolitain, 25
machine à machine, 28
maillé, 6
personnel, 25
point-à-point, 27
réparti, 6
rate based flow control, 41
raw.c
l.30–31, 401
RAW4_HTABLE_SIZE, 401
raw_v4_htable[], 401
raw_v4_input(), 401

rcu, 212
rcu_dereference(), 386
rcu_head (struct), 212
rcu_read_lock(), 316
rcu_read_unlock(), 316
rcupdate.h
  l.157–193, 316
  l.225–239, 386
  l.45–53, 212
read(), 69
read_lock(), 137
read_unlock(), 137
read_write.c
  l.81–84, 479
readv(), 86
recv(), 79
recvfrom(), 81
recvmsg(), 88
redirection, 105
register_filesystem, 467
register_netdev(), 233
register_netdevice(), 241
release_sock(), 416
remote
  connexion, 60
  socket, 66
  telephone, 66
renifleur, 142
représentation
  canonique, 10
requête
  de redirection, 357
  FIB, 357
request_irq(), 129
resynchronisation, 38
RFC, 57
  768, 111
  791, 108
  1071, 113
  1122, 10, 381
  1141, 113
  1310, 58
  1518, 107
  1519, 107
  1624, 113
  1700, 62, 110, 216
  1889, 60
  2292, 90
  2474, 109
rhash_entries, 368
Roberts, Larry, 13, 15, 43
rose/, 119
routage, 5, 9, 102, 105
  dynamique, 324
  statique, 324
  strict, 111
  table de, 14
  tolérant, 111
route, 9, 27
route.c
  l.113, 371
  l.131–133, 372
  l.1369–1376, 394
  l.1413–1453, 580
  l.148–159, 366
  l.1546–1814, 388
  l.1816–1881, 384
  l.1887-2178, 573
  l.197–202, 368
  l.203, 370
  l.204, 371
  l.205, 370
  l.207, 371
  l.212–216, 372
  l.2180-2212, 572
  l.2215–2230, 572
  l.2696, 368
  l.2706–2810, 368
  l.452–456, 375
  l.585–625, 345
  l.769–774, 375
  l.776–900, 373
  l.939–958, 584
route.h
  l.107–108, 386
  l.131–135, 554
  l.137, 386
  l.50–76, 367
  l.6–17, 367
  l.86–104, 371
router, 105
routeur, 5, 105
routing, 105
  loose, 111
rt_cache_flush(), 345
rt_cache_stat, 371
rt_cache_stat (struct), 371
RT_CACHE_STAT_INC(), 386
rt_check_expire(), 372
rt_drop(), 375

rt_flush_lock, 345
rt_flush_timer, 372
rt_hash_bucket (struct), 368
rt_hash_code(), 372
rt_hash_log, 371
rt_hash_mask, 370
rt_hash_rnd, 370
rt_hash_table[], 368
rt_intern_hash(), 373
rt_periodic_timer, 372
rt_run_flush(), 372
RT_SCOPE_HOST, 338
RT_SCOPE_LINK, 338
RT_SCOPE_NOWHERE, 338
RT_SCOPE_SITE, 338
RT_SCOPE_UNIVERSE, 338
rt_secret_rebuild(), 372
rt_secret_timer, 372
rt_set_nexthop(), 580
RT_TABLE_DEFAULT, 334
RT_TABLE_LOCAL, 334
RT_TABLE_MAIN, 334
RT_TABLE_MAX, 334
RT_TABLE_MIN, 332
RT_TABLE_UNSPEC, 334
RT_TOS(), 552
RTA_ALIGNTO(), 354
rta_cacheinfo (struct), 336
RTA_DATA(), 354
RTA_LENGTH(), 354
RTA_NEXT(), 354
RTA_OK(), 354
RTA_PAYLOAD(), 354
rta_session (struct), 337
RTA_SPACE(), 354
rtable (struct), 367
rtattr (struct), 336
RTAX_ADVMSS, 332
RTAX_MAX, 332
RTAX_MTU, 332
RTAX_RTT, 332
RTAX_WINDOW, 332
rtmsg (struct), 337
rtmsg_fib(), 347
RTN_ANYCAST, 338
RTN_BLACKHOLE, 338
RTN_BROADCAST, 338
RTN_LOCAL, 338
RTN_MAX, 338
RTN_MULTICAST, 338
RTN_NAT, 338
RTN_PROHIBIT, 338
RTN_THROW, 338
RTN_UNICAST, 338
RTN_UNREACHABLE, 338
RTN_UNSPEC, 338
RTN_XRESOLVE, 338
rtnetlink.c
    l.81, 235
rtnetlink.h
    l.105–113, 340
    l.108–120, 354
    l.125–142, 337
    l.144–165, 338
    l.168–190, 337
    l.192-211, 338
    l.220–231, 334
    l.295–309, 336
    l.311–344, 332
    l.33–68, 339
    l.350–368, 337
    l.812–819, 243
    l.821–826, 186
    l.96–106, 336
rtnl, 235
rtnl_lock(), 234
rtnl_sem, 234
rtnl_shlock(), 235
rtnl_shunlock(), 235
rtnl_unlock(), 235
RTO_ONLINK, 573
RTP, 60
RTPROT_BOOT, 337
RTPROT_KERNEL, 337
RTPROT_REDIRECT, 337
RTPROT_STATIC, 337
RTPROT_UNSPEC, 337
RW_LOCK_UNLOCKED, 137
rwlock_init(), 137
RWLOCK_MAGIC, 136
rwlock_t, 136
RX_BUF_CLR, 269
RX_CMD, 269
RX_GOOD, 270
RX_HIGH, 269
RX_LOW, 269
RX_MISSED, 270
RX_MULT, 270
RX_NORM, 270
RX_PROM, 270

RX_RUNT, 270
RX_STATUS, 269
rxrpc/, 119

s16, 123
s32, 123
s64, 123
s8, 123
sémaphore, 138
sa_family_t, 72
SA_INTERRUPT, 129, 130
SA_SAMPLE
    RANDOM, 129
SA_SHIRQ, 129
SAP, 12
SAPROM, 269
saut, 103
sch_generic.c
    l.185–206, 609
    l.208–213, 249
    l.215–230, 285
    l.232–238, 614
    l.240–250, 595
    l.252–256, 600
    l.258–265, 600
    l.267–284, 248
    l.286–303, 285
    l.306–330, 595
    l.332–347, 600
    l.349–360, 601
    l.399–447, 284
    l.40–63, 247
    l.449–457, 614
    l.515–549, 283
    l.551–569, 614
    l.571–580, 247
    l.582–599, 617
    l.65–69, 248
    l.82–183, 596
sch_generic.h
    l.27–56, 211
    l.58–107, 213
sched.h
    l.180, 418
sched/, 118
schedule_timeout(), 613
SCM_CREDENTIALS, 90
SCM_RIGHTS, 90
SCP, 61
scripts/, 118
scrutation, 80
sctp/, 119
SDU, 12
security.c
    l.27, 491
security.h
    l.1018–1020, 491
    l.1203–1233, 491
    l.650–778, 491
    l.92–94, 491
security/, 118
security_operations (struct), 491
security_ops, 491
security_sk_free(), 416
security_socket_bind(), 510
security_socket_create(), 488
security_socket_post_create(), 489
security_socket_recvmsg(), 525
security_socket_sendmsg(), 544
security_socket_shutdown(), 501
segment, 58
semaphore, 138
semaphore.h
    l.44–48, 138
send(), 79
sendmsg(), 88
sendto(), 81
serveur, 70
session layer, 10
set_bit(), 132
SET_MODULE_OWNER(), 273
setsockopt(), 91, 408, 410
SFD, 102
SFTP, 61
sget(), 472
SHUT_RD, 84
SHUT_RDWR, 84
SHUT_WR, 84
simple_statfs(), 475
size_t, 125
sk_add_node(), 517
sk_alloc(), 411
sk_alloc_slab(), 455
sk_alloc_slab_error(), 455
sk_buff (struct), 149
sk_buff_head (struct), 157
sk_cachep, 411
sk_common_release(), 499
sk_dst_check(), 553
sk_dst_get(), 553

sk_dst_reset(), 505
sk_dst_set(), 553
sk_dst_set.h
    l.950–966, 553
sk_filter (struct), 410
sk_filter_len(), 415
sk_filter_release(), 414
sk_for_each(), 435
sk_free(), 414
sk_head(), 401
sk_init(), 412
SK_RMEM_MAX, 412
sk_set_owner(), 499
sk_unhashed(), 517
SK_WMEM_MAX, 412
skb_append(), 179
skb_bond(), 316
skb_checksum(), 426
skb_checksum_help(), 594
skb_clone(), 181
skb_clone_fraglist(), 194
skb_cloned(), 170
skb_copy(), 183
skb_copy_and_csum_datagram_iovec(), 535
skb_copy_bits(), 185
skb_copy_datagram_iovec(), 533
skb_copy_expand(), 189
skb_cow(), 196
SKB_DATA_ALIGN(), 162
skb_dequeue(), 175
skb_dequeue_tail(), 176
skb_drop_fraglist(), 163
skb_frag_struct (struct), 148
skb_frag_t, 148
skb_free_datagram(), 537
skb_get(), 170
skb_headlen(), 186
skb_headroom(), 167
skb_init(), 181
skb_insert(), 178
skb_is_nonlinear(), 166
SKB_LINEAR_ASSERT(), 168
skb_over_panic(), 168
skb_pagelen(), 628
skb_peek(), 180
skb_peek_tail(), 180
skb_pull(), 169
skb_push(), 168
skb_put(), 167
skb_queue_empty(), 173
skb_queue_head(), 173
skb_queue_head_init(), 172
skb_queue_len(), 173
skb_queue_purge(), 177
skb_queue_tail(), 174
skb_queue_walk(), 568
skb_realloc_headroom(), 194
skb_recv_datagram(), 529
skb_release_data(), 163
skb_reserve(), 167
skb_set_owner_w(), 588
skb_share_check(), 171
skb_shared(), 170
skb_shared_info (struct), 148
skb_shinfo(), 162
skb_tailroom(), 166
skb_trim(), 195
skb_under_panic(), 169
skb_unlink(), 179
skb_unshare(), 171
skbuff.c
    l.1–39, 160
    l.110–164, 161
    l.1181–1199, 175
    l.1201–1218, 176
    l.1220–1233, 177
    l.1235–1253, 173
    l.1255–1273, 174
    l.1274–1298, 179
    l.1301–1318, 179
    l.1321–1338, 178
    l.1428–1437, 181
    l.167–178, 163
    l.180–183, 194
    l.188–203, 163
    l.205–212, 162
    l.214–257, 164
    l.259–348, 181
    l.350–399, 187
    l.401–441, 183
    l.444–497, 188
    l.499–559, 193
    l.561–579, 194
    l.582–636, 189
    l.667–714, 195
    l.71, 161
    l.716–853, 191
    l.79–92, 168
    l.855–930, 185
    l.94–108, 169

l.998–1070, 426
skbuff.h
l.1–15, 149
l.1055–1071, 186
l.1073–1077, 568
l.115–122, 157
l.1151–1158, 400
l.126–127, 149
l.129–135, 148
l.137–147, 148
l.149–283, 149
l.307, 166
l.313–314, 162
l.316–325, 173
l.327–338, 170
l.340–356, 164
l.36–38, 155
l.365–376, 170
l.378–388, 170
l.390–412, 171
l.40–41, 162
l.414–443, 171
l.445–464, 180
l.466–485, 180
l.48–71, 155
l.487–496, 173
l.498–503, 172
l.505–535, 174
l.537–560, 175
l.563–589, 176
l.592–605, 178
l.607–614, 179
l.616–632, 177
l.637–652, 177
l.655–658, 166
l.660–663, 186
l.665–672, 628
l.687, 168
l.701–719, 167
l.728–744, 168
l.73–84, 220
l.746–766, 169
l.768–791, 190
l.793–802, 167
l.804–813, 166
l.815–827, 167
l.853–892, 195
l.910–924, 178
l.926–962, 197
l.964–986, 196
skbuff_head_cache, 161
slab, 126
slab.c
l.2406–2426, 125
SLAB_CACHE_DMA, 127
SLAB_HWCACHE_ALIGN, 127
SLAB_NO_REAP, 127
sliding window, 40
SLIP, 59
slot, 43
SMP, 128
SMTP, 61
SNA, 13
snmp.h
l.1–10, 380
l.122–123, 380
l.71–75, 380
l.97–101, 430
snmp.h/linux
l.10–44, 380
SO_BROADCAST, 91
SO_DONTROUTE, 410
SO_ERROR, 91
SO_KEEPALIVE, 91
SO_LINGER, 91, 410
SO_OOBINLINE, 91
SO_PEERCRED, 410
SO_PRIORITY, 410
SO_RCVBUF, 91
SO_RCVLOWAT, 410
SO_RCVTIMEO, 410
SO_REUSEADDR, 91
SO_SNDBUF, 91
SO_SNDTIMEO, 410
SO_TYPE, 91
sock (struct), 404
sock.c
l.1153–1203, 416
l.1205–1213, 418
l.1217–1226, 416
l.1268–1280, 525
l.1297–1334, 499
l.130–144, 412
l.1338–1345, 455
l.178–184, 415
l.606–642, 411
l.644–667, 414
l.669–685, 412
l.737–750, 560
l.824–908, 561

l.84–88, 413
l.910–914, 561
l.916–930, 418
l.932–950, 416
sock.h
l.1–80, 357
l.10238–1244, 533
l.1050–1056, 588
l.1070–1108, 438
l.1125–1133, 530
l.115-267, 404
l.1223–1226, 531
l.1228–1231, 419
l.1246–1259, 528
l.269–280, 401
l.288–291, 517
l.318–327, 413
l.349–358, 517
l.371–372, 435
l.401–409, 415
l.496–567, 447
l.529–541, 418
l.572–575, 455
l.577–592, 499
l.594–598, 517
l.608–609, 513
l.615–618, 505
l.620–632, 525
l.634–637, 525
l.644-647, 476
l.649–652, 480
l.654–657, 495
l.73–82, 408
l.847–863, 414
l.871–894, 413
l.896–901, 414
l.903–917, 500
l.92-113, 406
l.937–948, 553
l.968–984, 505
l.999–1010, 553
sock_alloc(), 490
sock_alloc_inode(), 476
sock_alloc_send_pskb(), 561
sock_alloc_send_skb(), 561
SOCK_ASYNC_NOSPACE, 469
SOCK_ASYNC_WAITDATA, 469
SOCK_BINDADDR_LOCK, 505
SOCK_BINDPORT_LOCK, 505
sock_close(), 482
sock_common (struct), 406
sock_common_recvmsg(), 525
sock_create(), 487
sock_destroy_inode(), 477
SOCK_DGRAM, 68, 444
sock_disable_timestamp(), 415
sock_error(), 530
sock_fasync(), 481
sock_filter (struct), 410
sock_flag(), 415
sock_fs_type, 470
sock_hold(), 413
sock_init(), 449, 470
sock_init_data(), 416
SOCK_INODE(), 495
sock_inode_cachep, 476
sock_intr_errno(), 533
sock_iocb (struct), 525
sock_lock_init(), 413
sock_map_fd(), 494
SOCK_MAX, 444
sock_mnt, 490
sock_no_mmap(), 480
sock_no_open(), 481
SOCK_NOSPACE, 469
sock_orphan(), 500
SOCK_PACKET, 68, 444
sock_poll(), 479
sock_prot_inc_use(), 517
sock_put(), 414
sock_queue_rcv_skb(), 438
SOCK_RAW, 68, 444
sock_rcvtimeo(), 531
SOCK_RDM, 68, 444
sock_recv_timestamp(), 528
sock_recvmsg(), 522
sock_register(), 450
sock_release(), 483
sock_reset_flag(), 415
sock_sendmsg(), 543
SOCK_SEQPACKET, 68, 444
sock_sndtimeo(), 419
SOCK_STREAM, 68, 444
sock_unregister(), 451
sock_vfree(), 408
sock_wamalloc(), 560
sockaddr (struct), 72
sockaddr_ax25 (struct), 73
sockaddr_in (struct), 72
sockaddr_in6 (struct), 73

sockaddr_ipx (struct), 73
sockaddr_un (struct), 72
sockect.c
  l.274, 476
socket, 66
  buffer, 147
  client, 62, 71
  distante, 66
  locale, 66
  numéro, 67
  remote, 66
  serveur, 62, 70
  structure, 468
socket (struct), 468
socket(), 71
socket.c
  l.1–59, 119
  l.1055–1059, 480
  l.1074–1180, 487
  l.1187–1207, 486
  l.1275–1302, 510
  l.139–143, 444
  l.145–164, 451
  l.1499–1542, 542
  l.1553–1593, 521
  l.166–169, 451
  l.1663–1684, 500
  l.171–187, 489
  l.190–194, 490
  l.1984–2008, 450
  l.2010–2027, 451
  l.2032–2070, 449
  l.210–230, 510
  l.232–270, 523
  l.272, 470
  l.321–325, 475
  l.327–331, 471
  l.333, 490
  l.335–339, 470
  l.340–343, 477
  l.344–346, 477
  l.348–412, 494
  l.414–452, 501
  l.454–481, 490
  l.483–492, 481
  l.499–528, 483
  l.530–546, 543
  l.548–560, 543
  l.580–597, 524
  l.599–612, 522
  l.61–137, 477
  l.932–942, 479
  l.951–966, 482
  l.968–1045, 481
socket.h
  l.148–152, 339
  l.154–216, 442
  l.47–61, 520
  l.63–73, 520
socket_alloc (struct), 476
socket_file_ops, 477
SOCKET_I(), 480
socket_lock_t, 408
socket_state, 469
socketpair(), 67
sockets_in_use, 490
sockfd_lookup(), 501
sockfs, 470
sockfs_delete_dentry(), 477
sockfs_dentry_operations (struct), 477
sockfs_get_sb(), 471
SOCKFS_MAGIC, 470
sockfs_ops, 475
sofnet_data (struct), 208
softnet_data (struct), 300
SOL_SOCKET, 89, 91
SOL_TCP, 91
somme de contrôle, 36
  IP, 103
sound/, 118
sous-paquet, 109
sous-réseau, 25
sous-système
  réseau, v
Space.c, 229
  l.106–109, 261
  l.111–124, 260
  l.165–259, 261
  l.315–333, 258
  l.389–421, 230
  l.423, 230
spin_is_locked(), 136
spin_lock(), 135
spin_lock_bh(), 136
spin_lock_irq(), 136
spin_lock_irqsave(), 135
SPIN_LOCK_UNLOCKED, 135
spin_trylock(), 136
spin_unlock(), 136
spin_unlock_bh(), 136

spin_unlock_irq(), 136
spin_unlock_irqrestore(), 136
spin_unlock_wait(), 136
spinlock.h
  l.154–171, 136
  l.173–181, 137
  l.89–97, 135
  l.98–108, 135
spinlock_init(), 135
spinlock_t, 135
SS_CONNECTED, 469
SS_CONNECTING, 469
SS_DISCONNECTING, 469
SS_FREE, 469
SS_UNCONNECTED, 469
SSH, 60
ssize_t, 125
standard
  DIX, 45
start bit, 23
statfs (struct), 463
statfs.h, 475
statfs.h/asm-generic
  l.1–21, 463
stddef.h
  l.13–18, 568
stop bit, 23
store-and-forward, 5
Strowger, Almon, 4
SUD, 12
suffixe
  de protocole, 7
suite, 7
  de protocoles, 2
  TCP/IP, 8
sunrpc/, 119
super-bloc, 461
super.c
  l.213–260, 474
  l.264–314, 472
  l.622–630, 474
super_block (struct), 461
super_operations (struct), 462
support physique, 2
suseseconds_t, 127
synchronize_net(), 453
sys-fs.c
  l.412–456, 251
sys.c
  l.149–180, 240
sys_bind(), 510
sys_close(), 506
sys_recvfrom(), 521
sys_sendto(), 542
sys_shutdown(), 500
sys_socket(), 486
sysctl_ip_nonlocal_bind, 511
sysctl_ipfrag_high_thresh, 634
sysctl_ipfrag_low_thresh, 634
sysctl_local_port_range[], 514
sysctl_net_ipv4.c
  l.57, 498
sysctl_rmem_default, 412
sysctl_rmem_max, 412
sysctl_wmem_default, 412
sysctl_wmem_max, 412
sysfs_create_group(), 253
système
  informatique, 2
  ouvert, 3
  propriétaire, 3
système de fichiers
  virtuel, 460

tailroom, 148
tampon
  de socket, 147
TASK_INTERRUPTIBLE, 613
TASK_UNINTERRUPTIBLE, 613
tasklet, 131
tasklet_schedule(), 131
TASKLET_SOFTIRQ, 131
tasklet_struct (struct), 131
taux de transmission, 23
Taylor, Robert, 15
TC_PRIO_MAX, 595
TCP, 60
  numéro de protocole IP, 110
tcp.h
  l.59–73, 407
TCP/IP, 8
  architecture, 55
  implémentation, 55
  modèle, 55
TCP_CLOSE, 407
TCP_CLOSE_WAIT, 407
TCP_CLOSING, 407
TCP_ESTABLISHED, 407
TCP_FIN_WAIT1, 407
TCP_FIN_WAIT2, 407

tcp_ipv4.c
  l.99–104, 514
TCP_LAST_ACK, 407
TCP_LISTEN, 407
TCP_MAX_STATES, 407
tcp_protocol, 449
TCP_SYN_RECV, 407
TCP_SYN_SENT, 407
TCP_TIME_WAIT, 407
TCQ_F_BUILTIN, 211
TCQ_F_INGRES, 211
TCQ_F_THROTTLED, 211
TDM, 42
technique
  de remplissage, 39
technologie
  de transmission, 26
telnet, 60
terminaison, 29
test_and_change_bit(), 132
test_and_clear_bit(), 132
test_and_set_bit(), 132
test_bit(), 132
Thin Ethernet, 28
time out, 40
time-sharing, 13
time.h
  l.18–21, 127
time_t, 127
timer.c
  l.1087–113, 613
timer.h
  l.11–22, 128
  l.35–47, 128
timer_list (struct), 128
timeval (struct), 127
TLI, 65
token bus, 48
Token Ring, 59
token ring, 48
top half, 130
topologie
  d'un réseau, 27
TOS, 108
tr_configure(), 261
tr_type_trans(), 298
trailer, 36
trame, 36, 58
  Ethernet
    802.3, 102
    DIX, 100
transceiver, 29, 46
transcepteur, 29, 46
transfert
  de fichier, 61
transmetteur/récepteur, 29
transmission
  différée, 5
  filaire, 22
  sans fil, 22
transport layer, 9
try_module_get(), 489
TTL, 103
TX_16COLLISIONS, 270
TX_CMD, 269
TX_COLLISION, 270
TX_NORM, 270
TX_RDY, 270
TX_READY, 270
TX_STATUS, 269
type
  de paquet, 304
type de servive, 108
types.h
  l.1–5, 124
  l.1–50, 123
  l.13–16, 124
  l.29, 127
  l.54–70, 125
  l.77–80, 127
  l.92–124, 124

u16, 123
u32, 123
u64, 123
u8, 123
u_char, 123
u_int, 123
u_int32_t, 123
u_int64_t, 123
u_int8_t, 123
u_int_16_t, 123
u_long, 123
u_short, 123
UCB, 56
ucred (struct), 90, 339
ud_mib (struct), 430
UDP, 60
  numéro de protocole IP, 110
udp.c

l.1085–1108, 431
l.1110–1207, 428
l.117, 435
l.118, 433
l.120–202, 515
l.1353–1371, 503
l.209–217, 505
l.219–264, 434
l.266–276, 433
l.382–394, 569
l.396–472, 566
l.475–478, 432
l.480–668, 548
l.757–760, 438
l.768–862, 526
l.865–885, 504
l.978–1041, 436
udp.h
l.32, 435
l.43–52, 437, 516
l.50–55, 551
l.54–61, 568
l.54–67, 437
l.6–27, 421
l.77–80, 430
udp_check(), 432
udp_checksum_complete(), 431
udp_checksum_init(), 431
UDP_CSUM_DEFAULT, 568
UDP_CSUM_NORCV, 568
UDP_CSUM_NOXMIT, 568
udp_disconnect(), 504
udp_flush_pending_frames(), 569
udp_hash[], 435
udp_hash_lock, 433
UDP_HTABLE_SIZE, 435
UDP_INC_STATS(), 430
UDP_INC_STATS_BH(), 430
UDP_INC_STATS_USER(), 430
udp_lport_inuse(), 516
udp_opt (struct), 437
udp_port_rover, 515, 516
udp_prot, 503
udp_protocol, 398, 449
udp_push_pending_frames(), 566
udp_queue_rcv_skb(), 436
udp_rcv(), 428
udp_recvmsg(), 526
udp_sendmsg(), 548
udp_sk(), 437
udp_sock (struct), 437
udp_statistics, 430
udp_v4_get_port(), 515
udp_v4_lookup(), 433
udp_v4_lookup_longway(), 434
udp_v4_mcast_deliver(), 430
udp_v4_unhash(), 505
udphdr (struct), 421
uint, 123
uint16_t, 123
uint32_t, 123
uint64_t, 123
uint8_t, 123
uio.h
l.17–24, 520
l.7–9, 520
ulong, 123
un.h, 72
unchar, 123
unicast, 27, 60
unistd.h, 69
unité
de fragmentation, 110
unix/, 119
unlikely(), 166
unregister_netdev(), 615
unregister_netdevice(), 615
up(), 138
ushort, 123
usr/, 118
utilitaire
de Berkeley, 56
UUCP, 55

van Kempen, Fred, 145
verrou rotatif, 134
version, 277
VFS, 460
vlan/, 119
VTAM, 13

wait.c
l.78–94, 532
wait.h
l.325–332, 532
l.51–55, 408
wait_for_packet(), 531
wait_on_sync_kiocb(), 523
wait_queue_head_t, 408
WAN, 25

wanrouter/, 119
WARN_ON(), 166
web, 61
weight_p, 302
Wi-Fi, 25, 48
wild address, 73
write(), 69
write_lock(), 137
writev(), 86

X.25, 59
x25/, 119
xfrm.h
    l.262-284, 409
xfrm/, 119
xfrm_policy (struct), 409
XNS, 55

ZERONET(), 393
zone
    critique, 132