

Tartalomjegyzék

Előszó a második kiadáshoz	xv
1. Bevezetés	1
1.1. A Linux	1
1.2. A szabad szoftver és a Linux története	2
1.2.1. FSF	2
1.2.2. GPL	3
1.2.3. GNU	4
1.2.4. Linux-kernel	4
1.2.5. A Linux rendszer	6
1.2.6. Linux-disztribúciók	6
1.3. Információforrások	8
2. Betekintés a Linux-kernelbe	11
2.1. A Linux-kernel felépítése	11
2.2. A Linux elindulása	13
2.3. Processzek	14
2.3.1. A Linux-processzekhez kapcsolódó információk	15
2.3.2. A processz állapotai	17
2.3.3. Azonosítók	18
2.3.4. Processzek létrehozása és terminálása	19
2.3.5. A programok futtatása	20
2.3.6. Ütemezés	21
2.3.6.1. Klasszikus ütemezés	21
2.3.6.2. Az O(1) ütemezés	23
2.3.6.3. Teljesen igazságos ütemező	24
2.3.6.4. Multiprocesszoros ütemezés	25
2.3.7. Megszakításkezelés	25
2.3.8. Valós idejűség	26
2.3.9. Idő és időzítők	26
2.4. Memóriakezelés	27
2.4.1. A virtuálmemória-kezelés	27
2.4.2. Lapozás	28

2.4.3. A lapozás implementációja a Linuxon	29
2.4.4. Igény szerinti lapozás	31
2.4.5. Lapcsere	32
2.4.6. Megosztott virtuális memória	34
2.4.7. Másolás íráskor (COW technika)	34
2.4.8. A hozzáférés vezérlése	35
2.4.9. A lapkezelés gyorsítása	35
2.5. A virtuális állományrendszer	36
2.5.1. Az állományabsztrakció.....	36
2.5.2. Speciális állományok	38
2.5.2.1. Eszközállományok	38
2.5.2.2. Könyvtár	40
2.5.2.3. Szimbolikus hivatkozás.....	40
2.5.2.4. Csővezeték	40
2.5.2.5. Socket	41
2.5.3. Az inode	41
2.5.4. Az állományleírók	44
2.6. A Linux programozási felülete.....	44
3. Programkönyvtárak készítése	47
3.1 . Statikus programkönyvtárak.....	47
3.2. Megosztott programkönyvtárak	55
3.2.1. Megosztott programkönyvtár készítése.....	56
3.2.2. Megosztott programkönyvtárak használata	60
3.2.3. Megosztott programkönyvtárak dinamikus betöltése	63
3.3. Megosztott könyvtárak C++ nyelven	69
3.3.1. Programkönyvtárbeli C++-osztályok használata.....	69
3.3.2. C++-objektumok dinamikus betöltése programkönyvtárból	72
3.4. A megosztott könyvtárak működési mechanizmusai	77
3.4.1. A betöltött program	78
3.4.2. Statikus könyvtárat tartalmazó program linkelése és betöltése	80
3.4.3. Megosztott könyvtár linkelése és betöltése	83
3.4.3.1. A címtartomány kezelése	83
3.4.3.2. A megosztott könyvtárak megvalósításának alapkoncepciói.....	85
3.4.3.3. A megosztott könyvtárakkal kapcsolatos linkelés és betöltés.....	87
3.4.3.4. Dinamikusan linkelt megosztott könyvtár linkelése és betöltése	89
3.4.4. A programkönyvtárak használatának optimalizálása	89

4. Állomány- és I/O kezelés	95
4.1. Egyszerű állománykezelés	95
4.1.1. Az állományleíró	96
4.1.2. Állományok megnyitása	97
4.1.3. Állományok bezárása	98
4.1.4. Írás, olvasás és pozicionálás az állományban	99
4.1.5. Részleges és teljes olvasás.....	101
4.1.6. Az írásművelet finomhangolása.....	103
4.1.7. Állományok rövidítése	106
4.1.8. Állományok átirányítása	106
4.2. Inode-információk	108
4.2.1. Inode-információk lekérdezése.....	109
4.2.2. Jogok lekérdezése	110
4.2.3. Jogok állítása	111
4.2.4. Tulajdonos és csoport beállítása	112
4.2.5. Az időbélyeg beállítása	112
4.3. További állományműveletek.....	113
4.3.1. Eszközállományok és <i>pipe</i> bejegyzések létrehozása	113
4.3.2. Merev hivatkozás létrehozása.....	114
4.3.3. Szimbolikus hivatkozás létrehozása.....	115
4.3.4. Állományok törlése	116
4.3.5. Állományok átnevezése	116
4.4. Könyvtárműveletek	117
4.5. Csővezetékek	120
4.5.1. Névtelen csővezetékek.....	121
4.5.2. Megnevezett csővezetékek.....	123
4.6. Blokkolt és nem blokkolt I/O	126
4.7. A multiplexelt I/O módszerei	129
4.7.1. Multiplexelés a <i>select()</i> függvénnyel	129
4.7.2. Multiplexelés a <i>poll()</i> függvénnyel.....	134
4.7.3. A multiplexelési módszerek összehasonlítása.....	139
4.8. Állományok leképezése a memóriába.....	139
4.9. Állományzárolás	143
4.9.1. Zárolóállományok	144
4.9.2. Rekordzárolás	145
4.9.3. Kötelező zárolás	148
4.10. Kapcsolat a magas szintű állománykezeléssel	149
4.11. Soros kommunikáció	150
4.11.1. Kanonikus feldolgozás.....	151
4.11.2. Nem kanonikus feldolgozás.....	154

5. Párhuzamos programozás	157
5.1. Processzek.....	157
5.2. Processzek közötti kommunikáció (IPC).....	165
5.2.1. Szemaforok.....	166
5.2.2. Üzenetsorok	178
5.2.3. Megosztott memória	185
5.3. Processzek a Linux rendszerben	189
5.3.1. Feladatvezérlés	191
5.3.2. Démonok.....	193
5.3.3. Programok indítása <i>shell</i> ből.....	197
5.3.4. Jogosultságok	198
5.3.5. Felhasználói nevek és csoportnevek	200
5.4. Szálak	202
5.4.1. Szálak létrehozása	203
5.4.2. Szálak létrehozása C++ nyelven	207
5.4.3. Szálak attribútumai	210
5.4.4. Szálbiztos függvények	212
5.4.5. Szál leállítása.....	217
5.4.6. Szálak és a <i>fork/exec</i> hívások	220
5.5. POSIX-szinkronizáció.....	221
5.5.1. Kölcsönös kizárás (<i>mutex</i>)	221
5.5.2. Feltételes változók	227
5.5.3. Szemaforok.....	233
5.5.4. <i>Spinlock</i>	236
5.5.5. További lehetőségek: POSIX megosztott memória és üzenetsorok	238
5.6. Jelzések.....	238
5.6.1. A jelzéseküldés és -fogadás folyamata.....	239
5.6.2. Jelzések megvalósítása.....	245
5.6.3. A jelzéskezelő és a főprogram egymásra hatása	247
5.6.4. Jelzések és a többszálú processz	250
5.6.5. Jelzések programozása	250
5.6.5.1. Jelzések küldése	251
5.6.5.2. Jelzések letiltása és engedélyezése. Függőben lévő jelzések	253
5.6.5.3. A jelzések kezelése	254
5.6.5.4. Szinkron jelzéskezelés.....	258
5.6.6. A <i>SIGCHLD</i> jelzés	262

6. Hálózati kommunikáció	265
6.1. A socket.....	265
6.2. Az összeköttetés-alapú kommunikáció	267
6.2.1. A kapcsolat felépítése	268
6.2.2. A socket címhez kötése	268
6.2.3. Várakozás a kapcsolódásra	269
6.2.4. Kapcsolódás a szerverhez	269
6.2.5. A kommunikáció	270
6.2.6. A kapcsolat bontása	270
6.2.7. További kapcsolatok kezelése a szerverben	271
6.3. Az összeköttetés nélküli kommunikáció.....	272
6.3.1. A kommunikáció	273
6.3.2. A <i>connect()</i> használata	274
6.3.3. A socket lezárása	275
6.4. Unix domain socket	275
6.4.1. Unix domain socket címek	275
6.4.2. Unix domain socket adatfolyam szerveralkalmazás	276
6.4.3. Unix domain socket adatfolyam kliensalkalmazás.....	278
6.4.4. Unix domain socket datagram kommunikáció.....	280
6.4.5. Névtelen Unix domain socket	280
6.4.6. A Linux absztrakt névtére	280
6.5. IP	282
6.5.1. Röviden az IP-hálózatokról	282
6.5.2. Az IP protokoll rétegződése	284
6.5.3. IPv4-es címezés	285
6.5.4. IPv4-es címosztályok	286
6.5.5. IPv4-es speciális címek	287
6.5.6. IPv6-os címezés	288
6.5.7. Portok	289
6.5.8. A hardverfügő különbségek feloldása	290
6.5.9. A socketcím megadása	290
6.5.10. Lokális cím megadása	293
6.5.11. Név- és címfeloldás	294
6.5.11.1. A <i>getaddrinfo()</i> függvény.....	294
6.5.11.2. A <i>getnameinfo()</i> függvény.....	298
6.5.12. Összeköttetés-alapú kommunikáció	302
6.5.12.1. TCP kliens-szerver példa	304
6.5.12.2. TCP szerver alkalmazás	308
6.5.12.3. TCP-kliensalkalmazás	315
6.5.13. Összeköttetés nélküli kommunikáció	317
6.5.13.1. UDP-kommunikáció-példa	317
6.5.13.2. Többes küldés	320

6.6. Socketbeállítások	326
6.7. Segédprogramok.....	330
6.8. Távoli eljáráshívás	331
6.8.1. Az RPC-modell	332
6.8.2. Verziók és számok.....	332
6.8.3. Portmap	333
6.8.4. Szállítás.....	333
6.8.5. XDR	333
6.8.6. <i>rpcinfo</i>	334
6.8.7. <i>rpcgen</i>	334
6.8.8. Helyi eljárás átalakítása távoli eljárássá.....	335
7. Fejlesztés a Linux-kernelben.....	339
7.1. Verziófüggőség.....	340
7.2. A kernel- és az alkalmazásfejlesztés eltérései	341
7.2.1. Felhasználói üzemmód — kernelüzemmód	342
7.3. Kernelmodulok	343
7.3.1. Hello modul világ	344
7.3.2. Fordítás	346
7.3.3. A modulok betöltése és eltávolítása.....	347
7.3.3.1. <i>insmod/rmmod</i>	347
7.3.3.2. <i>modprobe</i>	347
7.3.4. Egymásra épülő modulok	348
7.4. Paraméterátadás a modulok számára	351
7.5. Karakteres eszközvezérlő.....	353
7.5.1. Fő- és mellékazonosító (major és minor number)	354
7.5.2. Az eszközállományok dinamikus létrehozása	355
7.5.3. Állományműveletek	356
7.5.4. Használatszámológó.....	357
7.5.5. „Hello világ” driver	358
7.5.6. Az <i>open</i> és a <i>release</i> függvények.....	362
7.5.7. A mellékazonosító (minor number) használata	363
7.5.8. Az <i>ioctl()</i> implementációja	366
7.6. A <i>/proc</i> állományrendszer.....	370
7.7. A hibakeresés módszerei	375
7.7.1. A <i>printk()</i> használata.....	375
7.7.2. A <i>/proc</i> használata	376
7.7.3. Kernelopciók	377
7.7.4. Az Oops üzenet.....	378
7.7.4.1. Az „Oops” üzenet értelmezése kernel esetében.....	379
7.7.4.2. Az „Oops” üzenet értelmezése kernelmodul esetében.....	380

7.7.5. Magic SysRq.....	381
7.7.6. A <i>gdb</i> program használata	382
7.7.7. A <i>kgdb</i> használata	382
7.7.8. További hibakeresési módszerek	383
7.8. Memóriakezelés a kernelben	384
7.8.1. Címtípusok	384
7.8.2. Memóriaallokáció.....	384
7.9. A párhuzamosság kezelése	386
7.9.1. Atomi műveletek.....	387
7.9.2. Ciklikus zárolás (spinlock)	389
7.9.3. Szemafor (semaphore)	391
7.9.4. Mutex.....	392
7.9.5. Olvasó/író ciklikus zárolás (spinlock) és szemafor (semaphore).....	393
7.9.6. A nagy kernelzárolás	394
7.10. I/O műveletek blokkolása	395
7.10.1. Elaltatás	396
7.10.2. Felébresztés.....	397
7.10.3. Példa	398
7.11. A <i>select()</i> és a <i>poll()</i> támogatása.....	403
7.12. Az <i>mmap</i> támogatása.....	404
7.13. I/O portok kezelése.....	407
7.14. I/O memória kezelése.....	408
7.15. Megszakításkezelés.....	410
7.15.1. Megszakítások megosztása	412
7.15.2. A megszakításkezelő függvények megkötései.....	412
7.15.3. A megszakítás tiltása és engedélyezése	412
7.15.4. A szoftvermegszakítás	413
7.15.5. A BH-mechanizmus	414
7.15.5.1. A kifeladat (tasklet)	414
7.15.5.2. Munkasor	417
7.16. A kernelszálak.....	421
7.17. Várakozás	423
7.17.1. Rövid várakozások	423
7.17.2. Hosszú várakozás	424
7.18. Időzítők.....	425
7.19. Eszközvezérlő modell	426
7.19.1. A busz	426
7.19.2. Eszköz- és eszközvezérlő lista	428
7.19.3. <i>sysfs</i>	429
7.19.4. Buszattribútumok exportálása	429

7.19.5. Az eszközevezérlő	430
7.19.6. Eszközevezérlő attribútumok exportálása	433
7.19.7. Az eszköz	433
7.19.8. Az eszköz attribútumainak exportálása	435
7.19.9. Példa	436
7.20. További információk	438
8. A Qt keretrendszer programozása.....	439
8.1. Az X Window rendszer.....	439
8.1.1. Az X Window rendszer felépítése	439
8.1.2. X Windows kliensalkalmazások	440
8.1.3. Asztali környezet	441
8.2. Fejlesztés Qt alatt.....	442
8.2.1. Hello Világ!.....	443
8.2.2. Projektállományok	447
8.2.3. A <i>QObject</i> szolgáltatásai.....	448
8.2.4. A <i>QtCore</i> modul.....	449
8.3. A Qt eseménykezelés-modellje	450
8.3.1. Szignálok létrehozása	452
8.3.2. Szlotfüggvények létrehozása	454
8.3.3. Szignálok és szlotok összekapcsolása	455
8.3.4. Szlot az átmeneti objektumokban.....	457
8.3.5. A Meta Object Compiler	458
8.4. Ablakok és vezérlők	459
8.4.1. Dialógusablakok készítése	460
8.4.2. A Qt vezérlőkészlete	466
8.4.3. Saját alkalmazásablakok	467
8.4.4. A főablak programozása	469
8.4.5. Lokalizáció	479
8.4.6. Saját vezérlők készítése.....	483
8.5. A dokumentum/nézet architektúra.....	489
8.5.1. Az alkalmazás szerepe.....	491
8.5.2. A dokumentumosztály	493
8.5.3. A nézetosztályok	497
8.5.4. További osztályok	503
8.6. További technológiák.....	509
8.6.1. Többszálú alkalmazásfejlesztés	509
8.6.2. Adatbáziskezelés.....	516
8.6.3. Hálózati kommunikáció.....	521
8.7. Összefoglalás	530

A függelék: Fejlesztőeszközök	533
A.1. Szövegszerkesztők	533
A.1.1. <i>Emacs</i>	533
A.1.2. <i>vi (vim)</i>	534
A.1.3. <i>nano (pico)</i>	534
A.1.4. <i>joe</i>	534
A.1.5. <i>mc</i>	534
A.1.6. Grafikus szövegszerkesztők	535
A.2. Fordítók	535
A.2.1. GNU Compiler Collection	536
A.2.1. <i>gcc</i>	536
A.2.3. LLVM	540
A.3. Make	541
A.3.1. Megjegyzések	542
A.3.2. Explicit szabályok	542
A.3.3. Hamis tárgy	544
A.3.4. Változódefiníciók	545
A.3.5. A változó értékadásának speciális esetei	546
A.3.6. Többsoros változók definiálása	547
A.3.7. A változó hivatkozásának speciális esetei	547
A.3.8. Automatikus változók	548
A.3.9. Többszörös cél	549
A.3.10. Mintaszabályok	550
A.3.11. Klasszikus ragozási szabályok	551
A.3.12. Implicit szabályok	552
A.3.13. Speciális tárgyak	553
A.3.14. Direktívák	554
A.4. Make alternatívák	554
A.4.1. <i>Autotools</i>	555
A.4.2. <i>CMake</i>	555
A.4.3. <i>qmake</i>	555
A.4.4. <i>SCons</i>	556
A.5. IDE	556
B függelék: Hibakeresés	557
B.1. gdb	557
B.1.1. Példa a <i>gdb</i> használatára	558
B.1.2. A <i>gdb</i> leggyakrabban használt parancsai	561
B.1.3. A <i>gdb</i> indítása	561
B.1.4. Töréspontok: <i>breakpoint</i> , <i>watchpoint</i> , <i>catchpoint</i>	562
B.1.5. Data Display Debugger (DDD)	566
B.1.6. Az IDE-k beépített hibakeresője	567

B.2. Memóriakezelési hibák	568
B.2.1. <i>Malloc</i> hibakeresők	569
B.2.1.1. Memóriaterület túlírása	570
B.2.1.2. Eléírás	571
B.2.1.3. Felszabadított terület használata	571
B.2.1.4. Memóriaszivárgás	571
B.2.1.5. A <i>malloc</i> hibakeresők korlátai	571
B.2.2. <i>Electric Fence</i>	572
B.2.2.1. Az <i>Electric Fence</i> használata	572
B.2.2.2. A <i>Memory Alignment</i> kapcsoló	574
B.2.2.3. Az eléírás	574
B.2.2.4. Az <i>Electric Fence</i> további lehetőségei	575
B.2.2.5. Erőforrásigények	575
B.2.3. <i>DUMA</i>	576
B.3. <i>Valgrind</i>	576
B.3.1. <i>Memcheck</i>	577
B.3.1.1. A <i>memcheck</i> modul működése	578
B.3.2. <i>Helgrind</i>	580
B.4. Rendszerhívások monitorozása: <i>strace</i>	582
B.5. Könyvtárfüggvényhívások monitorozása: <i>ltrace</i>	582
B.6. További hasznos segédeszközök	582
Tárgymutató	585
Irodalomjegyzék	593