

# Spis treści

Wstęp .....	17
1. Szybkie wprowadzenie do wbudowanego Linuksa .....	23
1.1. Architektura .....	24
1.2. Linux = Linux wbudowany .....	25
1.3. „Widok z helikoptera” .....	26
1.4. ARM .....	31
1.4.1. Cortex .....	33
1.4.2. OMAP .....	33
1.4.3. Operacje zmiennoprzecinkowe .....	33
1.4.4. XScale .....	34
2. Bardzo szybki start .....	35
2.1. Zanim zaczniemy .....	35
2.2. Cross-kompilator .....	36
2.3. Jądro – przygotowanie .....	37
2.4. Podstawowe narzędzia .....	38
2.5. Biblioteka standardowa C .....	39
2.6. Pliki urządzeń, skrypty startowe i konfiguracja .....	39
2.7. Ramdysk .....	42
2.7.1. Jądro – kompilacja .....	42
2.7.2. Uruchomienie systemu w emulatorze .....	42
2.8. Uruchomienie systemu na urządzeniu .....	43
3. Środowisko cross-kompilacji dla architektury ARM .....	45
3.1. Po co nam cross-kompilator? .....	46
3.2. Zanim zaczniemy: Decyzje projektowe .....	47
3.2.1. Nazwa toolchaina i jej znaczenie .....	48
3.2.2. (E)ABI .....	50

3.2.3.	Wersja procesora i architektury .....	52
3.2.4.	Zestawy instrukcji: ARM i Thumb .....	54
3.2.5.	Kolejność bajtów w słowie .....	55
3.2.6.	Operacje zmiennoprzecinkowe .....	56
3.2.7.	Obsługa wątków .....	57
3.3.	Zaczynamy: Przygotowanie systemu .....	58
3.3.1.	Konto robocze .....	58
3.3.2.	Środowisko .....	59
3.4.	Pliki nagłówkowe jądra .....	60
3.5.	GMP .....	61
3.6.	MPFR .....	61
3.7.	MPC .....	61
3.8.	Zlib .....	62
3.8.1.	Binutils .....	62
3.9.	GCC – wersja pomocnicza .....	65
3.10.	Biblioteka standardowa C .....	66
3.10.1.	uClibc .....	67
3.10.2.	eglibc .....	69
3.11.	libelf .....	74
3.12.	GCC – wersja docelowa .....	74
3.13.	Korzystanie z toolchaina .....	76
3.13.1.	Dodatkowe biblioteki kompilatora .....	76
3.13.2.	chrpath .....	76
3.13.3.	Usuwanie zbędnych symboli debugera .....	78
3.13.4.	Prosta weryfikacja działania kompilatora .....	78
3.14.	Co dalej? .....	79
4.	Narzędzia automatyczne .....	81
4.1.	Gotowe toolchainy .....	82
4.1.1.	CodeSourcery .....	83
4.1.2.	Linaro .....	85
4.2.	Crosstool-ng .....	85
4.2.1.	Instalacja .....	85
4.2.2.	Konfiguracja i budowanie toolchaina .....	86
4.2.3.	Minimalna wspierana wersja jądra .....	89
4.2.4.	Konfiguracja dla procesorów armv7 .....	91

---

4.2.5.	Toolchain prekonfigurowany .....	91
4.3.	Buildroot .....	92
4.3.1.	Instalacja .....	93
4.3.2.	Konfiguracja z nowym toolchainem .....	93
4.3.3.	Wybór programów i bibliotek .....	94
4.3.4.	Format docelowego systemu plików .....	96
4.3.5.	Buildroot z gotowym toolchainem .....	97
4.3.6.	Kompilacja .....	99
4.3.7.	Zmiana konfiguracji, przebudowanie lub usunięcie pojedynczego pakietu .....	99
4.3.8.	Zmienne środowiskowe .....	100
4.3.9.	Przeniesienie katalogu Buildroota .....	100
4.3.10.	Korzystanie z narzędzi .....	101
4.3.11.	Błędy w działaniu Buildroota .....	101
4.3.12.	Co dalej? .....	102
4.4.	OpenWRT .....	102
4.4.1.	Instalacja OpenWRT w wersji binarnej na urządzeniu .....	102
4.4.2.	Menedżer pakietów .....	105
4.4.3.	Obsługa Opkg .....	105
4.4.4.	Przykład: udostępnianie dysku USB dla systemów Windows .....	106
4.4.5.	Dodatkowe narzędzia – SDK .....	107
4.4.6.	Dodatkowe narzędzia – ImageBuilder .....	108
4.4.7.	Instalacja OpenWRT (z kodu źródłowego) .....	108
4.4.8.	Prototypowanie: OpenWRT w QEMU i na urządzeniu .....	111
4.5.	Scratchbox .....	114
4.5.1.	Scratchbox2 .....	115
4.5.2.	Scratchbox2 – Instalacja .....	115
4.5.3.	Scratchbox2 – Przygotowanie i konfiguracja .....	116
4.5.4.	Jak to działa? .....	117
4.5.5.	Uruchamianie i debugowanie programów bezpośrednio na urządzeniu .....	118
4.5.6.	Scratchbox (1) – instalacja i konfiguracja .....	121
4.5.7.	Jak korzystać z gotowych pakietów? .....	126
4.5.8.	Scratchbox (1) – Budowanie pakietów ze źródeł .....	130

---

4.6.	Inne narzędzia .....	131
4.6.1.	Emdebian .....	131
4.6.2.	Ptxdist .....	133
4.6.3.	LTIB .....	134
4.6.4.	MeeGo .....	135
4.6.5.	OpenEmbedded .....	137
4.7.	Instalacja systemu Debian 6.0 w wersji dla architektury ARM .....	139
4.7.1.	Instalacja – etap I (stacja robocza).....	140
4.7.2.	Instalacja – etap II (urządzenie).....	141
4.7.3.	Instalacja – etap III (podstawowa konfiguracja).....	142
5.	Emulatory .....	143
5.1.	QEMU .....	145
5.1.1.	Instalacja .....	145
5.1.2.	Emulowane platformy ARM .....	146
5.1.3.	Kompilacja QEMU ze źródeł .....	148
5.2.	Emulacja BeagleBoard, DevKit8000 i innych platform OMAP3 .....	150
5.2.1.	Typowe parametry wywołania .....	152
5.2.2.	Obrazy dysków i kart pamięci .....	153
5.2.3.	Obsługa sieci .....	154
5.3.	Obrazy płyt i dysków .....	156
5.4.	Emulacja urządzeń MTD .....	156
6.	Konfiguracja i kompilacja jądra .....	159
6.1.	Wersje Linuksa .....	159
6.1.1.	Rozwój jądra 2.6 .....	159
6.1.2.	Wersje eksperymentalne .....	161
6.1.3.	Dodatkowa dokumentacja .....	161
6.1.4.	Jak zdobyć kod źródłowy jądra .....	162
6.1.5.	Czy kompilować?.....	163
6.2.	Konfiguracja kodu źródłowego .....	164
6.2.1.	General set-up.....	167
6.2.2.	Enable loadable module support.....	167
6.2.3.	Enable the block layer .....	167
6.2.4.	System Type .....	168
6.2.5.	Processor type and features .....	169

---

6.2.6.	Boot options.....	169
6.2.7.	CPU Power Management.....	170
6.2.8.	Floating poin emulation.....	170
6.2.9.	Power management and ACPI options.....	170
6.2.10.	Bus options (PCI etc).....	170
6.2.11.	Executable file formats / Emulations.....	170
6.2.12.	Networking support.....	171
6.2.13.	Device Drivers.....	171
6.2.14.	Firmware Drivers.....	172
6.2.15.	File systems.....	172
6.2.16.	Kernel hacking.....	172
6.2.17.	Security options.....	173
6.2.18.	Cryptographic API.....	173
6.2.19.	Virtualization.....	173
6.2.20.	Library routines.....	173
6.2.21.	Kompilacja i instalacja.....	173
6.2.22.	Dodatkowa dokumentacja.....	174
7.	Jak działa jądro?.....	175
7.1.	Z czym się zmagamy?.....	175
7.2.	Sterownik urządzenia.....	176
7.3.	Przestrzeń użytkownika – procesy.....	177
7.3.1.	Cykl życia procesu, monitorowanie procesów.....	179
7.3.2.	Szeregowanie procesów.....	184
7.3.3.	Komunikacja między procesami.....	184
7.3.4.	Monitorowanie pamięci.....	185
7.4.	Interfejsy udostępniane przez jądro.....	187
7.4.1.	Wywołania systemowe.....	189
7.4.2.	System plików proc i inne oparte na pamięci RAM.....	190
7.4.3.	Pliki urządzeń.....	194
7.4.4.	udev.....	196
7.4.5.	mdev.....	198
7.4.6.	Pozostałe mechanizmy komunikacji.....	200
7.5.	Po stronie jądra.....	200
7.5.1.	Moduły.....	201

---

7.5.2.	Skąd wziąć narzędzia do zarządzania modułami? .....	202
7.5.3.	Jak ich używać .....	203
7.5.4.	Frameworki .....	204
7.5.5.	Model urządzeń w jądrze .....	205
7.5.6.	Magistrala .....	205
7.5.7.	Przykład: magistrala USB .....	206
7.5.8.	Specyfika systemów wbudowanych .....	208
7.5.9.	Przykład – port szeregowy w procesorach firmy Atmel .....	208
7.5.10.	Zasoby urządzeń .....	210
7.5.11.	Sterowniki – podsumowanie .....	211
7.6.	Obsługa USB .....	212
7.6.1.	Klient USB .....	213
7.6.2.	File-backed Storage Gadget .....	214
7.6.3.	Ethernet Gadget (with CDC Ethernet support) .....	215
7.6.4.	Serial Gadget (with CDC ACM and CDC OBEX support) .....	216
7.7.	GPIO w Linuksie .....	217
7.7.1.	LED Class Driver .....	218
7.7.2.	GPIO keys .....	219
7.7.3.	Ogólny sterownik GPIO .....	222
7.8.	Korzystanie z innych magistral .....	223
7.9.	Zarządzanie energią .....	224
7.9.1.	Liczniki .....	225
7.9.2.	Usypianie i przywracanie stanu systemu .....	226
7.9.3.	Interakcja z przestrzenią użytkownika .....	227
7.9.4.	Co robi procesor, gdy nic nie robi? .....	227
7.9.5.	Inne mechanizmy zarządzania energią .....	230
8.	Programowanie w jądrze .....	231
8.1.	Zanim zaczniemy .....	231
8.1.1.	Układ katalogów .....	231
8.1.2.	Język programowania i kompilator .....	232
8.1.3.	Brak biblioteki standardowej języka C .....	233
8.1.4.	Przenośność .....	234
8.1.5.	Rozmiar słowa .....	235
8.1.6.	Typy nieprzejrzyste .....	236

---

8.1.7.	Kolejność bajtów w słowie .....	236
8.1.8.	Typy o stałych rozmiarach .....	237
8.1.9.	Brak operacji zmiennoprzecinkowych .....	237
8.1.10.	Nie ma stałego API .....	237
8.1.11.	Zarządzanie pamięcią .....	238
8.1.12.	Synchronizacja jądra .....	241
8.1.13.	Obsługa przerwania .....	243
8.1.14.	Licencjonowanie jądra .....	246
8.1.15.	Prototypowanie sterownika .....	247
8.2.	Własne moduły jądra .....	248
8.2.1.	Niezbędne narzędzia i przygotowanie .....	248
8.2.2.	Kod źródłowy jądra .....	248
8.2.3.	Pierwszy moduł .....	249
8.2.4.	Kompilacja pierwszego modułu .....	250
8.2.5.	Dołączanie modułu do kodu jądra .....	251
8.2.6.	Funkcje udostępniane przez moduły .....	253
8.2.7.	Parametry modułów .....	255
8.2.8.	Parametry konfiguracyjne w /proc/sys .....	257
8.2.9.	Obsługa systemu plików debugfs .....	259
8.2.10.	Urządzenie znakowe .....	261
8.2.11.	Inne operacje na plikach – mmap .....	264
8.3.	Przykład z życia: sterownik portu szeregowego .....	268
8.3.1.	Implementacja przesyłania danych .....	271
8.3.2.	Wysyłanie oparte na próbkowaniu .....	272
8.3.3.	Wysyłanie oparte na przerwaniach .....	272
8.3.4.	Implementacja odbierania danych .....	272
8.3.5.	Linie kontroli modemu .....	274
8.3.6.	termios .....	275
8.3.7.	Konsola .....	275
9.	System bazowy .....	277
9.1.	Instalacja Busyboksa .....	278
9.2.	Ramdisk .....	281
9.3.	Urządzenie blokowe .....	282
9.3.1.	Systemy plików dla urządzeń blokowych .....	282

---

9.3.2.	Karty pamięci .....	284
9.3.3.	System plików squashfs .....	287
9.3.4.	FUSE i NTFS .....	287
9.4.	Pamięć Flash .....	288
9.4.1.	Opcje konfiguracji jądra .....	290
9.4.2.	Podział na partycje .....	293
9.4.3.	Systemy plików dla urządzeń MTD .....	296
9.4.4.	UBI .....	299
9.5.	Rozwiązania hybrydowe .....	302
9.6.	Sieciowy system plików .....	303
9.7.	XIP .....	306
9.8.	Obsługa systemów plików .....	307
9.8.1.	Montowanie .....	307
9.8.2.	Kontrola wykorzystania przestrzeni dyskowej .....	308
9.9.	Szyfrowane systemy plików .....	309
9.9.1.	ecryptfs .....	309
9.9.2.	device-mapper .....	310
9.10.	Praca z urządzeniem – narzędzia pomocnicze .....	311
9.10.1.	Kopia zapasowa .....	311
9.10.2.	Minicom .....	312
9.10.3.	Aktualizacja systemu na urządzeniu .....	315
10.	Uruchamianie systemu .....	317
10.1.	Bootstrap .....	318
10.1.1.	Uruchamianie systemu z karty SD .....	321
10.2.	SAM-BA .....	323
10.2.1.	RomBOT .....	323
10.2.2.	Wersje .....	327
10.2.3.	Korzystanie z programu .....	328
10.2.4.	SAM-BA i Linux .....	330
10.2.5.	Zmiany w kodzie appletów .....	331
10.3.	Toshiba AC100 (NVIDIA Tegra 2) .....	332
10.3.1.	Narzędzia deweloperskie .....	333
10.3.2.	Konfiguracja jądra i systemu bazowego .....	336
10.3.3.	Programowanie urządzenia .....	337
10.4.	Bootloader pierwszego poziomu .....	338



---

10.4.1.	X-loader .....	338
10.4.2.	AT91Bootstrap.....	339
10.4.3.	Freescale i.MX23 .....	339
10.4.4.	AT91 – własny program uruchomieniowy .....	342
10.4.5.	Parametry jądra .....	343
10.5.	u-boot .....	344
10.5.1.	Instalacja u-boota.....	345
10.5.2.	Pomoc .....	345
10.5.3.	Zmienne środowiskowe u-boota .....	346
10.5.4.	TFTP .....	347
10.5.5.	Instalacja i konfiguracja serwera tftpd .....	347
10.5.6.	mkimage .....	348
10.5.7.	Przykłady korzystania z u-boota .....	349
10.5.8.	Modyfikacja kodu źródłowego.....	349
10.6.	Jądro systemu .....	351
10.6.1.	Co się dokładnie dzieje w jądrze?.....	352
10.6.2.	Proces numer 1.....	355
10.7.	Optymalizacja czasu uruchamiania systemu .....	355
10.7.1.	Bootloader.....	356
10.7.2.	Jądro.....	356
10.7.3.	System bazowy .....	360
11.	Diagnostyka .....	365
11.1.	Sprzętowe narzędzia debugowania .....	366
11.1.1.	Oscyloskop .....	366
11.1.2.	JTAG .....	367
11.1.3.	Obsługa urządzeń JTAG .....	368
11.2.	Debugowanie aplikacji .....	369
11.2.1.	Instalacja .....	370
11.2.2.	Korzystanie z GDB .....	370
11.2.3.	Graficzne nakładki na GDB.....	374
11.3.	Śledzenie procesów .....	374
11.3.1.	Analizowanie działania skryptów .....	374
11.3.2.	Strace .....	375
11.3.3.	Śledzenie systemu .....	375

---

11.4.	Debugowanie jądra .....	376
11.4.1.	Opcje debugowania jądra .....	377
11.4.2.	Jak dobór kompilatora wpływa na jądro .....	378
11.4.3.	Wpisywanie komunikatów .....	379
11.4.4.	Poziomy rejestrowania .....	379
11.4.5.	Bufor komunikatów .....	380
11.4.6.	dev_info i dev_err .....	381
11.4.7.	Preprocesor C i przydatne makra .....	381
11.4.8.	Dedykowany system plików .....	383
11.4.9.	Oops .....	384
11.4.10.	Magic SysRq .....	384
11.4.11.	Debugowanie pamięci jądra .....	385
11.4.12.	Zastosowanie GDB do debugowania jądra .....	385
11.5.	KGDB .....	385
11.5.1.	Przygotowanie KGDB do pracy .....	386
11.6.	Debugowanie przy użyciu QEMU .....	388
11.6.1.	Debugowanie jądra przy pomocy QEMU .....	388
11.6.2.	Debugowanie U-BOOT-a .....	390
11.6.3.	Debugowanie uClibc .....	390
12.	Przykładowe projekty .....	393
12.1.	Skąd się bierze dystrybucja Linuksa? .....	393
12.2.	Konfiguracja wstępna .....	394
12.3.	Biblioteka ncurses .....	396
12.4.	PowerTOP .....	398
12.5.	Dźwięk w systemie Linux .....	399
12.5.1.	OSS .....	399
12.5.2.	ALSA .....	400
12.5.3.	alsa-lib .....	401
12.5.4.	SALSA .....	403
12.5.5.	Narzędzia ALSA .....	404
12.5.6.	Serwery dźwięku .....	405
12.5.7.	Portaudio .....	406
12.6.	Gstreamer .....	407
12.6.1.	Korzystanie z GStreamer .....	409

---

12.7. Mplayer .....	413
12.7.1. MPlayer i video4linux .....	414
12.8. Synteza mowy – rozwiązania opensource.....	416
12.8.1. Espeak .....	416
12.8.2. Festival – Flite .....	418
12.8.3. Komercyjne rozwiązania syntezy mowy .....	419
12.9. zlib .....	419
12.10. Kodowanie i dekodowanie dźwięku .....	420
12.10.1. Mp3 .....	421
12.10.2. Dekompresja mp3 .....	421
12.10.3. Kompresja mp3 .....	424
12.10.4. AAC/MP4.....	425
12.10.5. RealAudio .....	426
12.10.6. WMA .....	426
12.10.7. Ogg .....	426
12.11. Szyfrowanie .....	432
12.11.1. Szyfry symetryczne .....	434
12.11.2. Instalacja openssl .....	436
12.11.3. Instalacja crypted .....	436
12.11.4. Szyfry z kluczem jawnym .....	437
12.11.5. Podpis elektroniczny .....	438
12.11.6. Szyfrowanie haseł w Linuksie .....	439
12.11.7. Szyfrowanie z kluczem jawnym w praktyce .....	440
12.11.8. Własne CA, obsługa certyfikatów .....	442
12.11.9. Propozycje użycia .....	445
12.11.10. Wydajność szyfrowania.....	445
12.12. Serwer SSH .....	446
12.12.1. Uruchomienie serwera .....	447
12.12.2. Rozwiązywanie problemów .....	447
12.13. DirectFB.....	447
12.14. QT .....	452
12.14.1. Instalacja za pomocą Buildroota .....	452
12.14.2. Instalacja zindywidualizowana .....	453
12.14.3. Praca z biblioteką QT – GUI .....	455

12.15. QTOPIA .....	460
12.15.1. QT x86 .....	461
12.15.2. Biblioteki pomocnicze: uuid, tslib, jpeg .....	461
12.15.3. QT ARM .....	464
12.15.4. Przykładowa aplikacja .....	465
12.16. Maszyna wirtualna Javy (MIDP + CDC, CLDC) .....	466
12.16.1. Java w systemach wbudowanych .....	467
12.16.2. gcj .....	469
12.16.3. Biblioteka klas .....	469
12.16.4. JamVM .....	472
12.16.5. JamVM – testy .....	473
12.16.6. PhoneME .....	473
12.17. Obsługa systemów plików i urządzeń MTD .....	475
12.17.1. ext2, ext3, ext4 .....	475
12.17.2. LZO .....	476
12.17.3. MTD .....	478
12.18. Drukowanie .....	479
12.18.1. CUPS .....	479
12.18.2. Jak to działa? .....	481
12.18.3. Skąd wziąć sterowniki? .....	482
12.18.4. Instalacja komponentów systemu drukowania .....	483
12.19. Zegar czasu rzeczywistego .....	486
12.20. strace .....	488
12.21. Serwer HTTP .....	489
12.21.1. Sterowanie LED-ami .....	490
12.22. Sieci bezprzewodowe .....	491
12.22.1. Wprowadzenie w pigułce do 802.11 .....	492
12.22.2. Wi-Fi .....	492
12.22.3. Protokół .....	493
12.22.4. Bezpieczeństwo .....	493
12.22.5. Obsługa w systemie Linux .....	495
12.22.6. Przykładowe urządzenie .....	495
12.22.7. Interfejsy .....	500
12.22.8. Tryby pracy interfejsu .....	501

---

12.22.9. Sieć Ad-Hoc .....	502
12.22.10. Podłączanie do punktu dostępowego WPA2 .....	503
12.22.11. Konfiguracja punktu dostępowego .....	504
12.23. Zdalne sterowanie urządzeniem .....	505
12.23.1. VNC .....	505
12.23.2. X11vnc .....	506
12.23.3. LibVNCServer .....	507
12.23.4. Bufor ramki .....	508
12.23.5. Uinput .....	510
12.23.6. FbVNC .....	512
12.24. Przygotowanie systemu do uruchomienia .....	517
12.24.1. Konfiguracja .....	518
12.24.2. Przygotowanie .....	519
12.24.3. Kopiowanie plików i bibliotek .....	520
12.24.4. Haki .....	521
12.24.5. Skrypty startowe i pliki konfiguracyjne .....	522
12.24.6. Przygotowanie obrazu .....	523
A. Przygotowanie środowiska pracy .....	525
A.1. Wybór dystrybucji .....	525
A.2. Dobór sprzętu .....	527
A.2.1. Konfiguracja wirtualnej maszyny .....	528
A.3. Instalacja systemu Debian/GNU Linux .....	533
A.3.1. Skąd wziąć nośniki instalacyjne .....	533
A.3.2. Instalacja .....	534
A.4. Uwagi dotyczące użytkownika systemu .....	538
A.4.1. Konto administratora .....	538
A.4.2. Instalacja oprogramowania .....	539
B. Polecenia powłoki .....	541
B.1. Podstawowe narzędzia powłoki .....	541
B.1.1. Obsługa plików i katalogów .....	541
B.1.2. Wyświetlanie informacji o plikach .....	542
B.1.3. Wyświetlanie zawartości plików .....	542
B.1.4. Dopasowywanie nazw plików – znaki wieloznaczne .....	543
B.1.5. Przeszukiwanie plików .....	543

---

B.1.6.	Zmiana praw plików .....	543
B.1.7.	Porównywanie zawartości plików i katalogów .....	544
B.1.8.	Wyszukiwanie plików .....	544
B.1.9.	Przekierowania .....	544
B.1.10.	Kontrola procesów .....	545
B.1.11.	Rozmiary plików i partycji .....	545
B.1.12.	Archiwa .....	545
B.1.13.	Drukowanie .....	546
B.1.14.	Zarządzanie użytkownikami .....	546
B.1.15.	Zarządzanie czasem .....	546
B.1.16.	Pomoc .....	547
B.1.17.	Obliczenia .....	547
B.1.18.	Administracja .....	547
B.1.19.	Jądro, sprzęt, moduły .....	548
B.1.20.	Interfejsy jądra .....	548
B.1.21.	Praca zdalna .....	549
B.1.22.	HTTP .....	549
B.2.	Edytor tekstu VIM .....	549
B.3.	Skrypty powłoki .....	552
B.4.	Kompilator gcc .....	554
C.	Praca z kodem źródłowym, system kontroli wersji GIT .....	555
C.1.	Tworzenie łat .....	555
C.2.	Korzystanie z GIT-a .....	556
C.2.1.	Początek pracy .....	556
C.2.2.	Nowy projekt .....	556
C.2.3.	Pobranie istniejącego projektu .....	557
C.2.4.	Historia zmian .....	558
C.2.5.	Tworzenie łat .....	558
C.2.6.	Gałęzie projektu .....	558
C.2.7.	Praca grupowa .....	559
C.3.	Przeglądanie plików źródłowych .....	561
C.3.1.	Grep i wyrażenia regularne .....	561
C.3.2.	cscope .....	563
C.3.3.	LXR .....	564