

# Dobrodošli u Linux i Kylix

**Z**A PROGRAMERE KOJI DOLAZE IZ SVETA WINDOWS, LINUX JE POTPUNO NEISTRAŽENO OKRUŽENJE. KYLIX PREDSTAVLJA NOV NAČIN "ŽIVOTA" ZA PROGRAMERE KOJI RADE U LINUXU. OVO POGLAVLJE JE UVOD ZA PROGRAMERE. PRVO ĆEMO PREDSTAVITI KLJUČNE ELEMENTE LINUXA, A ZATIM RAZMATRATI INSTALIRANJE KYLIXA. NAKON TOGA ĆEMO NAPRAVITI NAŠU PRVU APLIKACIJU.

UVOD U LINUX NE MOŽE BITI OPŠIRAN, JER BI ZA TAKAV UVOD BILA POTREBNA CELA KNJIGA. UVOD JE NAMENJEN PROGRAMERIMA I BAVI SE DISKUSIJOM NEKOLIKO TEHNIČKIH RAZLIKA KOJE DOLAZE IZ OPERATIVNOG SISTEMA WINDOWS.

## Linux 101

Pre nego što se upustimo u razmatranje tehničkih elemenata, dopustite nam da Vam damo naš kratak uvod u Linux. Linux je besplatan operativni sistem koji se veoma brzo razvija i predstavlja san svakog programera, jer Vam daje moć kontrole nad svim funkcijama operativnog sistema, uključujući izvorni kod celog OS-a i većine aplikacija koje se izvršavaju.

Molimo Vas samo da Linux ne reklamirate previše:

- Nemojte govoriti da nema bagova, jer to nije tačno. Razlika između Windowsa i Linuxa je u postojanju ljudi koji su voljni da odmah isprave probleme umesto proizvođača koji žele da prodaju sledeću verziju koja će imati korisne osobine i možda nekoliko bagova manje (ili će možda imati drugačije bagove).
- Nemojte govoriti da je jednostavan, iako postaje sve lakši za korišćenje, a najnovije aplikacije (kao što je KDE-ov Konqueror koji vidite na slici 1.1) nisu ništa lošije od Windowsovog Explorera.
- Možete govoriti da je besplatan, što znači da imate slobodu (nešto kao slobodu govora), a ne da ne košta (kao besplatno pivo). Open source softver Vam nudi veći izbor, ali da biste ga efikasno koristili morate platiti usluge ili verzije koje su spremne za korišćenje. Verovatno ćete potrošiti manje novca nego kada odaberete neki drugi softver, ali, verovali ili ne, to nije najvažnija prednost.
- Možete govoriti da je stabilan, robustan i da Vam daje veću kontrolu i istovremeno kontroliše korisnika (korisničke privilegije su stvarnost).
- Možete govoriti da se razlikuje od drugih operativnih sistema. Windows je više prijateljski, ali je nepredvidljiv. Linux je teže savladati i konfigurisati, ali kada nešto radi, nastaviće da radi godinama.

Osnovna razlika je ta što se korisnici moraju priviči na Linux. Postojanje velikog broja programa koje možete koristiti za obavljanje istog posla, korišćenje komandne linije za operacije niskog nivoa, postojanje velike količine kvalitetnog besplatnog softvera -to je sve ono što nije uobičajeno u Windowsu. U Windowsu ne možete ni sanjati da uključite sopstvenu verziju radne površine i menadžera prozora, da zamenite grafički podsistem ili da birate između različitih kernela ili niza naprednih implementacija sistema datoteka - da ne pominjemo mogućnost menjanja sistemskog koda i pravljenje sopstvenih verzija.



**SLIKA 1.1** Linux KDE 2 radna površina na kojoj je otvoren pretraživač Konqueror

Sloboda koju imate u Linuxu je velika vrednost ne samo za idealiste, već i za profesionalne programere iz praktičnih razloga. To znači da možete proučiti Linuxov izvorni kod, koristiti ga kako Vam je volja i naučiti kako tačno radi (nešto što u Windowsu ne možete ni da zamislite). Linuxov izvorni kod možete kompajlirati i instalirati na hiljadama računara, koristiti samo neke njegove delove, unaprediti sistem samo kada je neophodno i tako dalje. Postoje hiljade open source aplikacija koje možete proučiti kako biste unapredili svoje znanje (ako imate kompletan izvorni kod tih aplikacija). Takođe, možete unaprediti izvorni kod postojećih aplikacija spajanjem open source projekta, dajući drugima kod kojim se ispravljaju bagovi i poboljšavaju karakteristike kojima ste se bavili.

Iako Vam ovo poglavlje ne može dati kompletan uvod u Linux, mi mislimo da вреди utrošiti nešto vremena rekapitulirajući ključne koncepte Linuxa i UNIX-a, koncepte koje morate razumeti pre nego što budete mogli uspešno da koristiti Kylix.

## Distribucije Linuxa

Postoje dva osnovna načina za formiranje Linux sistema: preuzimanje kernela i izvornog koda biblioteka i samostalno pravljenje sistema, ili preuzimanje gotovih distribucija (koje se često nazivaju distro). Postoje mnoge distribucije Linuxa, a najpopularnije su Red Hat, Suse, Mandrake, Caldera i Debian (možete pretražiti Web i pronaći daleko više informacija nego što ih mi možemo pružiti u ovoj knjizi). Većina distribucija ima besplatnu verziju sa smanjenim brojem funkcija koju možete preuzeti sa mreže ili dobiti uz časopise ili knjige, kao i jednu ili više proširenih verzija koje možete kupiti ili se pretplatiti pa dobijati unapređenja. Ako je potrebno da veoma brzo i efikasno instalirate Linux, preporučljivo je da instalirate potpunu verziju koja sadrži neki oblik profesionalne podrške.

Kasnije u ovom poglavlju pročitajte odeljak "Instaliranje Kylixa" kako biste saznali koje distribucije Borland zvanično preporučuje i koje su minimalne verzije koje su dovoljne za rad u Kylixu i izvršavanje Kylix aplikacija. Još je bolje proveriti odeljak o Kylixu na Borlandovom Web sajtu (adresa je <http://www.borland.com/kylix>).

## Bilo jednom davno

Linux je rođen kada je Linus Torvald (Linus Torvald) odlučio da izmeni Minix, klon Unixa za Intelove procesore koji je, vežbe radi, napravio profesor Endrju S. Tanenbaum (Andrew S. Tanenbaum). Pošto licenca Minixa Torvaldu nije dozvoljavala da napravi izmene, on je napisao sopstveni operativni sistem. Iako je mnogo toga od tada promenjeno, Linux je svakako operativni sistem iz Unix familije i svako ga može besplatno koristiti i menjati.

## "Sve što možete videti je ... prompt!"

Pošto instalirate Linux, prvo što ćete videti nakon podizanja sistema je prost zahtev za prijavljivanjem korisnika kako biste se identifikovali pre nego što pristupite sistemu. U najnovijim distribucijama možete se prijaviti u grafički sistem Linuxa. Mada ovakav način prijavljivanja može pomoći početnicima, mi ga ne preporučujemo programerima. Ako želite da postanete Linux programer, prvo morate razumeti kako funkcionišu korisnici i konzole. Pošto se priviknete na Linux, možete odabrati način prijavljivanja koji Vam najviše odgovara, ali mi verujemo da ćete nastaviti da koristite konzolu. To je jednostavno praktičnije.

Ako niste mladi, verovatno se sećate tekstualnih avanturističkih igara. Pre nego što je nastupilo vreme zadivljujuće grafike, sve što smo imali na raspolaganju su bile komandna linija i prosti tekstualni opisi. Zadavanjem tekstualnih komandi mogli smo da rešimo logičke zagonetke i da istražujemo nepoznate tamnice. To su bile zanimljive, ali frustrirajuće igre, jer ako nismo koristili prave reči ili niz reči za rešavanje zagonetke, nismo mogli napredovati kroz igru. Ponekad je jedino rešenje bilo nasumično korišćenje reči.

Korišćenje Linuxa sa konzole ponekad dovodi do istog stanja frustriranosti do kojeg su nas dovodile te stare igre. Možete se naći u situaciji da rešavate neki komplikovan posao koji ne možete da završite, ali ako ste uporni, u jednom trenutku ćete naći rešenje (pa ćete biti ponosni na sebe). Međutim, Vi, koji niste toliko mladi, nemojte misliti da je Linuxova konzola slična DOS promptu, jer je razlika u mogućnostima i složenosti ogromna. Zajedničko im je samo to što su to konzole komandne linije i kao takve se veoma razlikuju od grafičke radne površine na koju ste navikli.

Ovo može biti zabavno, ali nemaju svi korisnici vremena za veoma dugačak proces učenja. Međutim, veoma je važno znati kako se koristi menadžer datoteka i konzola. Korišćenjem menadžera datoteka brže ćete naučiti kako da radite sa datotekama, ali Vam to neće pomoći da razumete šta se dešava, a neki poslovi će se daleko duže obavljati. Ako utrošite vreme na učenje trikova na niskom nivou, bićete nagrađeni većom kontrolom nad poslovima, bržim operacijama i još nečim.

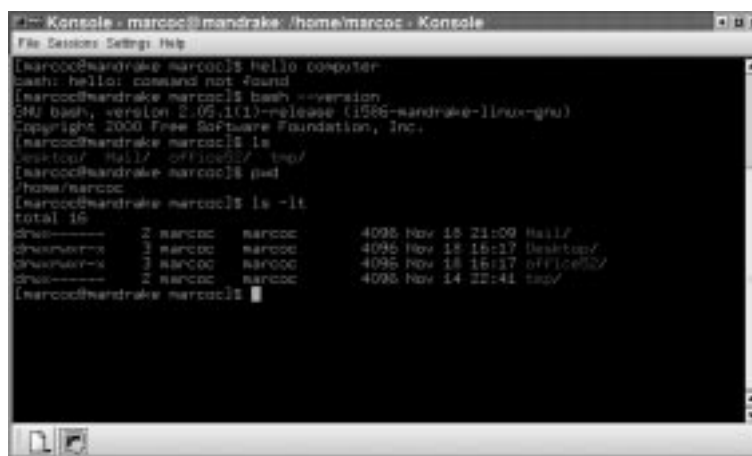
## Konzola bez granica

Ukolo su Vam se svidеле stare MS-DOS komande, bićete zadovoljni što je u Linuxu osnovni alat Bash (Bourne again shell - školjka Bourne) i što je to alat koji se u Linuxu najčešće koristi.

O korišćenju školjke Bourne postoji nekoliko odličnih knjiga, pa ćemo mi ovde samo objasniti neke od komandi koje se najčešće koriste kao i neke komande koje se ne koriste toliko često, ali su od velike pomoći programerima. Nakon tog uvoda trebalo bi da možete samostalno da radite i istražujete mogućnosti Linuxove školjke. Zapamtite da se sve što otkucate u konzoli preusmerava u školjku Bash koja će obaviti odgovarajuće akcije. Na slici 1.2 vidite primer Bash sesije koja obuhvata dve komande; komande ls i pwd, koje ćemo sada razmatrati.

### NAPOMENA

Komande koje se najčešće koriste su kratke. Koreni ovakvog pristupa vode iz vremena pre nego što su se koristili monitori, kada su standardni izlazni uredaji bili matrični štampači i kada su se ulazni podaci unosili pomoću bušenih kartica. Kratke komande danas smanjuju korišćenje tastature. Kada počnete da ih koristite, mogu Vam izgledati kao šifre. ■



```
[marcoc@mandrake marcoc]$ hello computer
bash: hello: command not found
[marcoc@mandrake marcoc]$ bash --version
GNU bash, version 2.05.1(1)-release (1996-mandrake-linux-gnu)
Copyright 2000 Free Software Foundation, Inc.
[marcoc@mandrake marcoc]$ ls
Desktop/ Mail/ office52/ tmp/
[marcoc@mandrake marcoc]$ pwd
/home/marcoc
[marcoc@mandrake marcoc]$ ls -lt
total 16
drwxr-xr-x 2 marcoc marcoc 4096 Nov 15 21:09 Mail/
drwxr-xr-x 3 marcoc marcoc 4096 Nov 15 16:17 Desktop/
drwxr-xr-x 3 marcoc marcoc 4096 Nov 15 16:17 office52/
drwxr-xr-x 2 marcoc marcoc 4096 Nov 14 22:41 tmp/
[marcoc@mandrake marcoc]$
```

SLIKA 1.2 Primer prozora Bash konzole kada je zadato nekoliko komandi

Na levoj strani svakog novog reda, školjka Bash prikazuje prompt koji je sličan sledećem promptu:

```
[marcoc@mandrake marcoc]$
```

Prvi element je ime tekućeg korisnika za kojim sledi znak @ i ime računara. Mi smo sigurni da ste upoznati sa takvim oblikom prompta: to je uobičajeni oblik elektronske adrese - sada znate odakle potiče! Iza imena računara se nalazi znak za razmak i ime tekućeg direktorijuma. Na kraju se prikazuje znak \$ koji korisnika obaveštava da nije administrator. Ako se prijavite kao administrator, umesto ovog znaka će se prikazati znak #. Prompt sadrži samo tekući direktorijum. Ako je nepohodno da se prikazuje cela putanja, otkucajte komandu pwd, što je skraćenica za print working directory (prikaži radni direktorijum).

Da biste prikazali datoteke tekućeg direktorijuma, upotrebite komandu `ls`. Njeno značenje je jednako komandi `dir` u DOS-u, pa za nju možete koristiti džoker znake `*` i `?`. Međutim, komanda `ls` ima daleko više opcija od komande `dir`. Otkucajte `ls --help` i prikazaće se opcije koje se najčešće koriste. Više informacija o datotekama možete dobiti ako zadate komandu `ls -l` (ili `ls -lt` kako biste datoteke prikazali tako da budu sortirane prema vremenu kada su menjane).

### SAVET

Parametrima komandne linije obično prethodi znak minus kada se koriste parametri dužine jednog slova (recimo, `lt -a`) ili dva znaka minus kada se koriste parametri koji su predstavljeni celom rečju (recimo, `lt --help`). ■

Kada zatražite detaljan spisak, prikazaće se mnogo više detalja nego u DOS-u ili Windowsu zbog razlika u sistemu datoteka. Na primer, možete prikazati sledeći red:

```
drwxrwxr-x  3 marcoc      marcoc  4096 Nov 18  16:17 Desktop/
```

Prvi znak odgovara tipu datoteke: vrednost može biti `d` ako je u pitanju direktorijum, `-` ako se radi o običnoj datoteci ili `l` ako je u pitanju simbolički link. Iza znaka kojim se određuje tip datoteke se nalaze dozvole datoteke (`rw-rwxr-x` u prethodnom redu) i broj linkova ka datoteci (uključujući stvarne i simboličke linkove, teme koje ćemo kasnije obraditi). Zatim sledi ime vlasnika i grupe (u našem primeru obe vrednosti su `marcoc`), veličina datoteke (`4096`), datum izmena i na kraju ime datoteke. U ovom i naredim odeljcima ćemo se pozabaviti mnogim drugim tehničkim detaljima (od prepoznavanja malih i velikih slova do dozvola).

## Jedan sistem, mnogo korisnika

Linux i UNIX sistemi su višekorisnički računari. Na svakom računaru postoji nalog za administratora koji se naziva `root` i jedini može konfigurisati računar, instalirati softver koji mogu koristiti svi korisnici i ostvariti pristup nekim sistemskim direktorijumima i komandama.

S obzirom na zaštitu koja postoji za obične korisnike, uobičajeno je korišćenje dva različita naloga (obični nalog i nalog administratora) čak i na računaru koji koristi jedan korisnik, uključujući prenosne računare. Zapravo, za svakodnevne poslove bolje je prijaviti se kao korisnik koji nema `root` nalog. Na taj način nećete moći da izmenite veći deo sistemskih podešavanja što sprečava nenamerno narušavanje sistema. Ako je neophodno da promenite podešavanja, onda se možete prijaviti kao administrator dok su ostali korisnici aktivni. U Linuxu je moguće da više korisnika bude istovremeno prijavljeno za različite aplikacije (ili različite instance konzole ili grafičkog menadžera datoteka).

Svaki korisnik ima svoj direktorijum, koji se obično nalazi unutar direktorijuma `/home`, sa izuzetkom administratora koji koristi direktorijum `/root`. Na osnovni direktorijum (`/home` direktorijum) tekućeg korisnika (recimo, `/home/marcoc`) upućuje promenljiva okruženja `HOME` i prečica `~`. Ove informacije možete koristiti u kodu da biste zadavali putanje za zapisivanje korisničkih podataka.

## Menjanje korisnika (i konzola)

U Linuxu se uvek možete izdavati za nekog drugog korisnika, a možete se i ponovo prijaviti i raditi kao dva ili više korisnika. Kada želite da promenite tekućeg korisnika, možete koristiti komandu su za kojom sledi željeno korisničko ime. Kada zadate komandu, od Vas će biti zatraženo da unesete lozinku. Ako u komandi ne zadate korisničko ime, podrazumeva se da želite da se prijavite kao administrator (naravno, ako znate lozinku). Da biste se vratili na prethodnog korisnika treba da zadate komandu `exit`.

Ako ste se prijavili kao administrator, od Vas se neće tražiti da unesete lozinke drugih korisnika jer ste administrator celog sistema i možete pristupiti informacijama drugih korisnika (uključujući i lozinke). Kao što je već istaknuto, dobra zaštita podrazumeva da se uvek prijavite kao običan korisnik i da se kao administrator prijavite (pomoću komande `su`) samo kada želite da instalirate nov softver ili obavite osetljive operacije.

Simultano možete otvoriti veliki broj konzola čime se simulira veliki broj terminala. Na svaku novu konzolu se morate prijaviti kao da radite sa udaljenog računara. Najjednostavniji način je korišćenje kombinacije tastera `Ctrl+Alt` i nekog od funkcijskih tastera `F1` do `F6`. Tasteri `F7` do `F12` su obično rezervisani za XWindows sesije. Pomoću kombinacije tastera `Ctrl+Alt+F1` ćete se vratiti na originalnu konzolu, a pomoću kombinacije tastera `Ctrl+Alt+F7` ćete se vratiti u XWindows.

Zbog toga su XWindows sesije (ili XSesije) veoma slične konzolama (ne uzimajući u obzir grafiku) iako se za te sesije nije neophodno prijaviti jer pripadaju tekućem korisniku. Komanda za pokretanje XWindows servera (ili XServera) je `xinit`. To je veoma složena komanda. Zbog toga svaka distribucija sadrži skript školjke kojim se pokreće X pri čemu se koriste unapred zadati parametri. Ime skripta je obično `startx`.

Između XWindows i konzola možete prelaziti bez ograničenja. Ipak, budite oprezni ako Vašem računaru mogu pristupiti drugi korisnici, jer bi tada trebalo da zatvorite sve konzole, naročito one u kojima ste prijavljeni kao administrator. Ako to ne uradite, neko može slučajno (ili namerno) izmeniti i uništiti Vaš sistem i konfiguraciju kada neko vreme ne nadgledate računar.

Kada postoji aktivna XSesija ostali korisnici ne mogu aktivirati XWindows, izuzev ako ne pokrenete još jedan XServer. Pošto danas retko ko na Linux sistemu ima grafički terminal, dodatne XSesije se retko koriste. Međutim, smatram da su takve sesije korisne prilikom debugovanja, naročito ako morate održavati stabilnu KDE sesiju dok testirate grafičke aplikacije.

### SAVET

XServeri se imenuju `:0`, `:1` i tako dalje. Drugi XServer pokrećete zadavanjem komande `startx --:1`. ■

## Dobijanje pomoći

Većina Linux distribucija sadrži obimnu dokumentaciju, ali ona nije integrisana kao Windowsova pomoć. To je jedina prednost Windowsa, a Windowsova tehnička dokumentacija (na nivou programera) je skoro potpuno nedostupna u programima namenjenim krajnjim korisnicima.

Kao što smo već pomenuli, Linuxove komande koje u potpunosti ne razumete (ili ih se ne možete setiti) možete zadati parametrima `-h` ili `--help`. Da biste dobili opširniju pomoć možete koristiti *man pages* (man se skraćena za *manual* - uputstvo; dakle, stranice uputstava), koje se odbijaju pomoću odgovarajuće komande. Takođe, zadatu reč možete tražiti u svim stranicama uputstva zadavanjem komande apropos. Pored komandi školjke Bash, stranice uputstva objašnjavaju mnoge programerske zadatke koji obuhvataju sve od Linuxovog kernela do sistema datoteka. Stranice uputstva su niskog nivoa, ali su zato prilično opširne. Informacione stranice (koje se pozivaju zadavanjem komande `info` za kojom sledi ime komande) sadrže hiprelinkove tako da lako možete prelaziti sa jednog na drugi dokument, ali nisu toliko opširne. Howtos su prvobitno napravljene kao tekstualne datoteke i predstavljaju još jedan izvor informacija koje se naročito odnose na instaliranje i otklanjanje grešaka.

Svaka distribucija Linuxa ovim standardnim sistemima pomoći dodaje sopstvenu dokumentaciju, pri čemu su tradicionalne info stranice i stranice uputstva formatirane kao HTML ili PDF datoteke, pa čak i u obliku baze podataka koja se može pretraživati. Iako je ovakav tip informacija prvenstveno namenjen krajnjim korisnicima (a ne programerima), verovatno ćete te informacije smatrati korisnim. U Vašem sistemskom meniju ili radnoj površini potražite link ka dokumentaciji.

---

**SAVET**

Kylixova pomoć obuhvata više izvora informacija, uključujući stranice uputstva, tako da kada zatražite informaciju o funkciji koristeći taster F1, možete dobiti Borlandov ili sistemski dokument. ■

## Sistem datoteka

Linuxov sistem datoteka ima malo sličnosti sa Windowsom i DOS-om pri čemu je jednak samo mali broj aspekata. DOS je zapravo izveden iz CP/M-a, jer se u to vreme smatralo da je UNIX previše složen za mikro-računare i prve PC-je (što je verovatno bilo tačno). U kasnijim verzijama DOS-a su uvedene neke komande i karakteristike UNIX-a. To je objašnjenje zašto su neke komande, kao što su `cd`, `echo` i `md`, jednake na oba sistema. U DOS-u su takođe pozajmljeni simboli za preusmeravanje i simbol `|` (pipe), kao što ćete videti kasnije u ovom poglavlju. Začudjujuće je da je separator putanje (kosa crta, `/`) u DOS-u zamenjena obrnutom kosom crtom (`\`). Kako je rad sa datotekama važan za krajnjeg korisnika i takođe od izuzetne važnosti za programe koje pravite, ovaj odeljak o osnovama Linuxa je nešto opširniji od ostalih odeljaka u ovoj knjizi.

---

**NAPOMENA**

Na mnogim sajtovima možete uporediti istorijat Linuxa i DOS-a, uključujući, za DOS, <http://www.maxframe.com/HISZMSD.HTM> i za Linux, <http://www.cab.u-szeged.hu/local/linux/linux-birth.html>. ■

Linuxov sistem datoteka razlikuje mala i velika slova. To je osnovna razlika koja prouzrokuje dosta problema korisnicima i programerima koji su navikli na Windowsov sistem datoteka koji ne razlikuje mala i velika slova. Postoji nekoliko Windowsovih tehnika (kao što je pretvaranje imena datoteka u velika slova pre poredenja) koje se uopšte ne mogu koristiti.



Formati datuma i vremena datoteka u Linuxu se razlikuju od formata koji se koristi u Windowsu (u Kylix postoje funkcije pomoću kojih možete konvertovati te vrednosti). Zapamtite da se vreme i datum datoteke menjaju kada je kopirate i da sistem pravilno beleži poslednje pristupanje datoteci. Zapravo, u Linuxu, kao i u novijim verzijama Windowsa, postoje zastavice za svaku datoteku, ali u odnosu na DOS postoje neke suptilne razlike. Kako biste bolje razumeli datume datoteka u Unixu zadajte komandu `info touch`.

## Drvo sistema datoteka

Linux/UNIX sistem datoteka ima hijerarhiju u kojoj postoji samo jedan koren, koji može obuhvatiti više fizičkih uređaja. Počinje se od uređaja koji je priključen kao koren (`root`), a zatim se ostali uređaji mogu priključiti pomoću *virtuelnih* poddirektorijuma. Početna tačka drveta (u obrnutom položaju) se naziva koren (engl. `root`) i obično se označava sa `Root` ili, tehnički, kosom crtom (`/`).

### NAPOMENA

Direktorijum `Root` i korisnik administrator imaju isto ime (koje se obično zapisuje tako da početno slovo bude veliko ili malo), ali su to dva potpuno različita koncepta. Prilično čudno. ■

Distribucije obično prave standardne direktorijume. Mi Vam savetujemo da napravite sve svoje direktorijume u okviru direktorijuma korisnika. Da biste dobili potpuno objašnjenje svih standardnih direktorijuma, zadajte komandu `man hier`.

Kao što je istaknuto, kao separator putanje koristi se kosa crta (`/`), za razliku od DOS-a i Windowsa gde se koristi obrnuta kosa crta (`\`). U Linuxu se obrnuta kosa crta (`\`) može koristiti za potpuno drugačiju svrhu: za nastavljavanje komande u drugom redu. Pokušajte da unesete sledeću komandu:

```
ls \  
-lta
```

U ovom primeru obrnuta kosa crta nema mnogo smisla, jer je komanda kratka, ali za dugačke komande je vrlo korisno nastaviti ih u novom redu, naročito u okviru skripta.

### NAPOMENA

Obrnuta kosa crta (`\`) se u Linuxu koristi kao escape simbol, što znači da se svaki znak koji se nalazi iza obrnute kose crte doslovno shvata, nezavisno od znaka za novi red koji se ignoriše. Obrnutu kosu crtu možete koristiti za prosleđivanje imena datoteke koje kao parametar sadrži razmake ili druge specijalne znake, recimo kao u sledećoj komandi `md new\ dir\ -with\ \strange\ \simbol\&`. Naravno, savetujemo Vam da se klonite korišćenja komplikovanih komandi! ■

U Linuxu ne postoje koncept uređaja i oznaka uređaja, jer se svi oni priključuju (pomoću komande `mount` ili prečice na radnoj površini). Takođe, za označavanje niza direktorijuma u putanji u Linuxu se koristi dvotačka (`:`), dok se u Windowsu koristi tačka-zarez (`;`).

**SAVET**

Da bi prevazišli razlike, Kylix i Delphi programeri koji žele da njihovi programi budu portabilni sa jednog u drugi operativni sistem, mogu koristiti konstantu `PathDelim` koja je definisana u jedinici `SysUtils`. Takođe postoji konstanta `DriveDelim` koja se dodeljuje samo u Windowsu kada je njena vrednost dvotačka (:) i konstanta `PathSep` čija vrednost može biti dvotačka (:) ili tačka-zarez(;). ■

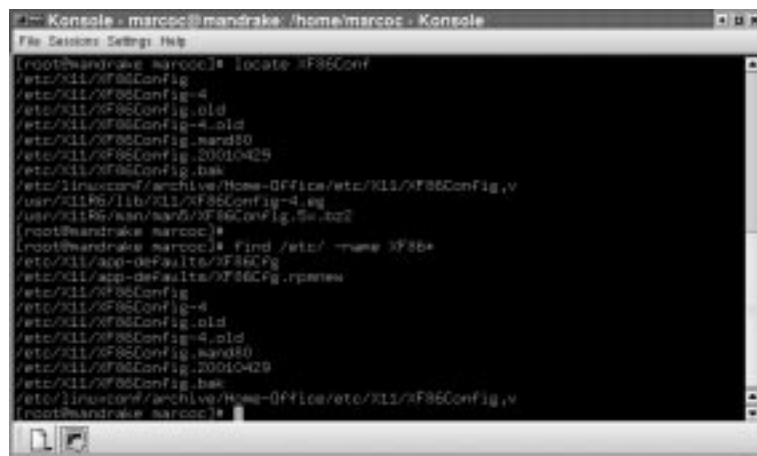
Druga velika razlika u formatima datoteka se odnosi na tekstualne datoteke. Tekstualne datoteke u Linuxu koriste samo znak Line Feed (znak čiji je kod u ASCII tabeli 10, a u Pascalu je to konstanta `#10`) za obeležavanje novog reda, umesto kombinacije znaka Enter i Line Feed (`#13` i `#10`) koja se koristi u Windowsu. Promenljive RTL-a `DefaultTextLineBreakStyle` i `sLineBreak` Vam omogućavaju da napišete kod koji se može preneti iz Delphija 6 u Kylix (više reči o ovim promenljivama u poglavlju 5).

**Kretanje kroz sistem datoteka**

Za kretanje kroz direktorijume sistema datoteka možete koristiti komandu `cd` kao što to činite u DOS-u. U školjci Bash postoji veoma moćan mehanizam za dovršavanje komadi: dovoljno je da pritisnete taster Tab na bilo kojoj poziciji u okviru putanje (ili imena datoteke), a školjka Bash će dovršiti putanju (ako postoji samo jedna mogućnost) ili će Vam prikazati moguće vrednosti. Morate probati da uradite ovo kako biste mogli da shvatite koncept, ali kada se naviknete, nećete moći bez njega da radite.

Ako ste instalirali miša, možete ga koristiti za kopiranje teksta u konzolama. Dovoljno je da selektujete rečenicu koja Vam je potrebna i da desnim tasterom miša kliknete mesto gde želite da je kopirate. U X-Terminalu za ovu operaciju možete koristiti kombinaciju tastera `Ctrl+Insert`. Ostali specijalni tasteri koje možete koristiti obuhvataju kursor tastere na gore i na dole, pomoću kojih prolazite kroz istorijat komandi koje ste koristili. Za svakog korisnika se održava zasebna lista komandi koja se obično zapisuje između dve sesije. Kombinacija tastera `Ctrl+Backspace` odgovara komandi `undo`, kojom se uklanja poslednji znak (slično kombinaciji tastera `Ctrl+Z` u Windowsu). Ako je potrebno da neki posao obavite u drugom direktorijumu, a želite da se kasnije vratite u tekući direktorijum, koristite komande `pushd` i `popd`.

Da biste pronašli datoteku, koristite komande `find` i `locate`. Komanda `locate` se mnogo brže izvršava jer koristi indeks. Međutim, ne možete je koristiti za skorije datoteke jer je potrebno obnoviti indeks, a to može uraditi samo administrator pomoću komande `updatedb`. U konzoli koju vidite na slici 1.3 prikazan je primer upotrebe ovih komandi.



**SLIKA 1.3** Korišćenje komandi *find* i *locate* u konzoli

Komanda *locate* pronalazi sve datoteke koje sadrže reč koju prosleđujete kao parametar. Komanda takođe omogućava pretraživanje bez razlikovanja malih i velikih slova zadavanjem parametra *-i*. Zbog indeksiranja koje smo pomenuli, komanda *locate* se obično koristi za pronalaženje sistemskih datoteka. Za pronalaženje korisničkih datoteka verovatnije je da ćete koristiti komandu *find*.

Sintaksa komande *find* je nešto komplikovanija: prvi parametar je direktorijum u kome počinje traženje, a za imenom se navodi *-name* i tačno ime datoteke (i verovatno džoker znaci). Linuxov sistem datoteka je zaista hijerarhijski, tako da pretraživanje koje počinje iz root direktorijuma, odnosno direktorijuma */*, zapravo znači da treba pretražiti sve diskove koji su priključeni, uključujući udaljene diskove i druge sisteme datoteka. Ukoliko se traženje oduži, možete upotrebiti kombinaciju tastera *Ctrl+Z* kako biste prekinuli tekuće traženje.

## Rad sa datotekama

Postoji još nekoliko komandi za rad sa datotekama koje vredi naučiti. Zapamtite da možete koristiti taster *Tab* za selektovanje datoteka ili putanja koje se kao parametri prosleđuju ovim komandama:

- Za kopiranje datoteka ili direktorijuma koristite komandu *cp*. Ova komanda radi isto kao i DOS-ova komanda *copy*.
- Za promenu imena ili premeštanje datoteka koristite komandu *mv*. (Ako komandu koristite za promenu imena datoteka onda to možete poistovetiti sa premeštanjem datoteke u isti direktorijum pri čemu joj se dodeljuje novo ime.)
- Za automatsku promenu imena većem broju datoteka pri čemu se koriste određena pravila, koristite komandu *rename*.

- Za uklanjanje datoteka (ali ne i direktorijuma) koristite komandu `rm` ili komandu `rm -f` ako ne želite da se od Vas traži potvrda o uklanjanju datoteka. Dobar haker može oporaviti uklonjene datoteke, pa ako ste paranoični možete koristiti komandu `shred`. Na taj način ćete sasvim sigurno onemogućiti oporavljanje datoteka (to niko neće moći da uradi, pa ni Vi!).
- Za pravljenje novog direktorijuma koristite komandu `md`. Za uklanjanje direktorijuma koristite komandu `rd`. Direktorijum mora biti prazan pre nego što budete mogli da ga uklonite.

**SAVET**

Može se desiti da imate problema sa datotekama koje počinju znakom `-`. Iako je korišćenje ovog znaka dozvoljeno i imenima datoteka, taj znak će za većinu komandi biti protumačen kao parametar. Da biste rešili ovaj i slične probleme, upotrebite simbol `--` kojim označavate da u komandnoj liniji nema više parametara. Mi smo ga koristili ranije: u skriptu `startx` smo koristili simbol `--` za razdvajanje parametara skripta `startx` od parametara za pokretanje XServera. ■

**Dozvole i ekstenzije**

U Linuxu svaka datoteka ima dozvole kojima se označava da li se datoteka može čitati, zapisivati (modifikovati) ili izvršavati, što je označeno slovima `r`, `w` ili `x`, respektivno, ili crticom ako ne postoji odgovarajuća dozvola. Postoji više skupova dozvola koji se navode jedan iza drugoga: prvi skup se odnosi na vlasnika datoteke, drugi za sve članove grupa korisnika kojima vlasnik pripada i treći skup za sve ostale korisnike. To znači da svaka datoteka ima devet zastavica pristupa, što Vam daje veliku fleksibilnost i kontrolu.

U sledećem redu vidite devet zastavica `rw-rw-r-x` u kopiji izlaza komande `ls` koju ste ranije već videli:

```
drwxrwxr-x  3 marcoc  marcoc      4096 Nov 18 16:17 Desktop/
```

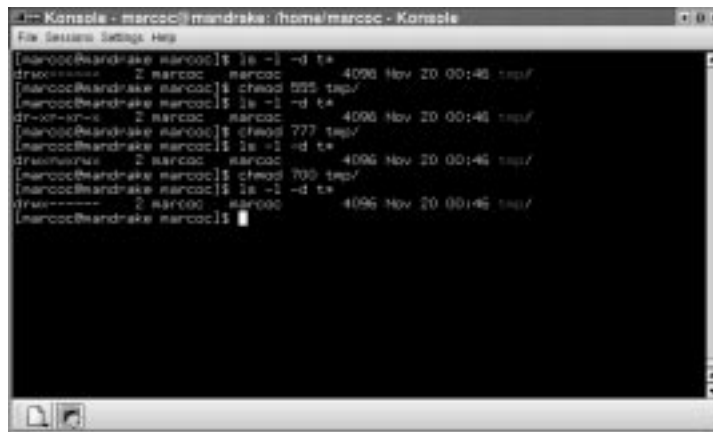
Ovaj direktorijum može biti čitan, zapisivan i izvršavan ako je korisnik `marcoc` i ako je iz grupe `marcoc`. Svaki drugi korisnik može samo čitati i izvršavati ga (kao i datoteke koje sadrži), ali ne može zapisivati u direktorijum ni iz njega uklanjati datoteke.

Da biste promenili dozvole, koristite komandu `chmod`. Programeri ovu komandu obično koriste u sledećem obliku (iako postoje drugi, napredniji načini njenog korišćenja):

```
chmod octal-mode filename
```

*Octal-mode* (oktalni režim) je trocifreni kod u kome svaka cifra označava dozvole za korisnika koji je vlasnik datoteka, korisnike koji pripadaju grupi vlasnika datoteke i sve ostale korisnike, respektivno.

Svaka cifra je oktalni broj koji se dobija dodavanjem bitova za svaku dozvolu: 4 za čitanje, 2 za zapisivanje i 1 za izvršavanje. Na primer, oktalni broj 777 znači da svi korisnici sa datotekom mogu raditi šta im je volja, a broj 555 znači da se datoteka može samo čitati, dok broj 700 znači da samo vlasnik datoteke može sa njom raditi šta mu je volja dok joj ostali korisnici ne mogu pristupiti. Više primera vidite na slici 1.4.

**SLIKA 1.4** Primeri dozvola i komande chmod

Ako ste pomalo zbunjeni, zapamtite da će Vam grafički menadžer datoteka u KDE-u prilično pomoći, kao što vidite na slici 1.5.

**SLIKA 1.5** Dozvole u KDE menadžeru datoteka

Dozvole datoteka donose prihod svakom sistemskom administratoru u Unixu, pa nije predviđeno da ih u potpunosti razumete. Kada se naviknete na njihove ezoterične osobine (kao što je sticky bit) ili počnete da razmišljate u oktalnim brojevima, onda možete reći da ste definitivno ušli u svet Unixa.

Jedan od elemenata Unixovih dozvola jeste to što su izvršne datoteke (uključujući batch datoteke) specifičnom zastavicom, pa im nije neophodna ekstenzija (za razliku od DOS-a i Windowsa). Ekstenzije datoteka su od manje važnosti u Linuxu, jer datoteke sadrže magičan broj kojim se određuje njihov tip. Međutim, pošto se *magični brojevi* odnose samo na najčešće tipove datoteka (zapravo, samo na nekoliko tipova), ekstenzije se često koriste za detaljnije određivanje tipa datoteka (na primer, za označavanje dnevnike datoteke kako bi se razlikovala od ostalih tekstualnih datoteka).

Menadžeri radnih površina, kao što je KDE, često koriste MIME tipove radi pridruživanja grupe ekstenzija ikoni i programu ili listi programa. Na ovaj način se može oponašati ponašanje Windowsove radne površine. Ako kliknete datoteku sa registrovanom ekstenzijom, KDE će pokrenuti aplikaciju koja je pridružena datoteci, a datoteku će proslediti kao parametar. MIME tipovi su originalno napravljeni za pridruživanje datoteka elektronskim porukama.

---

**NAPOMENA**

MIME je skraćenica za Multipurpose Internet Mail Extension i toliko je složena tema da je ovde nećemo objašnjavati, ali na Internetu možete naći obimnu dokumentaciju o MIME tipovima kao i info stranice i stranice uputstva. ■

Još jedna familija datoteka je predstavljena simboličkim linkovima. Po svom ponašanju ove datoteke podsećaju na Windowsove .lnk datoteke. Datoteke simboličkih linkova zapravo ne postoje. To su samo reference (odnosno linkovi) ka fizičkoj datoteci. Često se koriste kao da su preuzete iz Windowsa kako bi se pojednostavio pristup aplikacijama koje se često koriste, ali ih možete upotrebiti za mapiranje datoteka preko drugih datoteka.

Klasičan primer simboličkog linka su deljeni objekti (SO), Linuxova verzija dinamičkih biblioteka (DLL). Ovu temu ćemo razmatrati u poglavlju 13.

Simboličke linkove možete koristiti kada je aplikaciji potrebna datoteka iz nekog određenog direktorijuma; često iz istog direktorijuma u kome se nalazi aplikacija, ali se datoteka nalazi negde drugde u sistemu. Umesto da kopirate datoteku, možete napraviti simbolički link i otvoriti datoteku tako što ćete otvoriti datoteku simboličkog linka.

Još jedan način korišćenja simboličkih linkova, verovatno način koji ćete najčešće koristiti, jeste mogućnost brzog pristupa datoteci pravljenjem linka na radnoj površini korisnika.

---

**SAVET**

Savremene distribucije imaju direktorijum radne površine u okviru osnovnog (home) direktorijuma svakog korisnika. Datoteke ili linkovi koje smeštate u taj direktorijum se prikazuju na grafičkoj radnoj površini korisnika. ■

Komanda za pravljenje simboličkog linka je `ln -s`:

```
ln -s target-file link-name
```

Simbolički link se odnosi na ime datoteke i putanju ciljne datoteke, tako da će link biti prekinut ako preimenujete ili premestite datoteku. Pored toga, ako napravite datoteku čije ime je jednako ciljnoj datoteci linka, link će se odnositi na novu datoteku.

**NAPOMENA**

Simbolički linkovi su tako nazivaju jer Unixove datoteke sadrže linkove ka fizičkim datotekama koje se nazivaju inode. Ti linkovi se nazivaju hard linkovi kako bi se razlikovali od simboličkih linkova koji se često nazivaju sym-linkovi. U Linuxu, datoteku možete kopirati a da to ne utiče na podatke datoteke na disku, jer ste samo dodali još jedan link imenu inode. Fizička datoteka će biti uklonjena samo kada se uklone svi njeni hard linkovi. Kada napravite hard link datoteke, njeno editovanje će dovesti do promena drugih linkova jer se oni odnose na istu datoteku. Hard link možete napraviti komandama `ln` i `cp -l`, a tekuće brojeve inode možete prikazati komandom `ls -li`. To je komanda koju ćete retko koristiti, ali je korisno da je znate. ■

**Sakrivene i konfiguracijske datoteke**

Među Windowsovim atributima datoteka nalazi se zastavica Hidden (skrivena datoteka) koja ne postoji u Linuxu. Po konvenciji, skrivene datoteke i direktorijumi u Linux OS-u počinju tačkom (kao za direktorijum `~/.kde` u kojem KDE zapisuje svoje konfiguracijske datoteke). Da biste prikazali skrivene datoteke, zadajte komandu `ls -a` ili komandu `ls -la`. Većina menadžera datoteka imaju zastavicu pomoću koje možete prikazati skrivene datoteke. Ako uključite ovu zastavicu videćete mnoge skrivene datoteke u osnovnom direktorijumu svakog korisnika.

U Linuxu ne postoji centralna Registry datoteka u koju se zapisuju informacije o konfiguraciji. Međutim, postojanje jedne datoteke u koju se zapisuje gotovo sve što se odvija u sistemu je slaba tačka Windowsa tako da sada i Microsoft smanjuje korišćenje takve datoteke za podešavanje aplikacija. Linux koristi skrivene datoteke koje imaju root dozvole za sistemsko podešavanje i skrivene datoteke u osnovnom direktorijumu tekućeg korisnika ili u određenom poddirektorijumu radi podešavanja aplikacija. Različite Linux distribucije koriste različite direktorijume za sistemsko podešavanja. Na taj način imate veću fleksibilnost nego kada postoji monolitna kolekcija informacija. Dalje, opšte pravilo je podešavanje promenljive okruženja koja se odnosi na sve direktorijume aplikacija.

**SAVET**

Da biste iz Kylixa došli do podešavanja, možete koristiti klasu `TStringList` pri čemu koristite prednost parova ime/vrednost. Ovu klasu ćete upoznati u poglavlju 6. Ukoliko koristite podešavanja onda koristite format Windowsovih INI datoteka (parovi ime/vrednost koji su grupisani po odeljcima). Takođe možete iskoristiti Kylixove klase `TIniFile` ili `TMemIniFile` koje se mogu prevesti u Delphi za Windows. ■

**Još Linuxovih karakteristika**

Pošto smo razmotrili neke osnovne karakteristike Linuxa i sistema datoteka, možemo se pozabaviti drugim temama koje se odnose na procese, usmeravanje i skriptovanje. Ovim temama ćemo završiti naše veoma kratko objašnjenje osnova Linuxa.

**Procesi i tokovi**

Linux je nedvosmisleno višeprogramski operativni sistem, mnogo više nego što je to Windows (ili što je bar bio). Sve do nedavno u Linuxu nije posvećivano dovoljno pažnje radu u više tokova. Serverske aplikacije su često razdvajale procese (fork) umesto da pokrenu više tokova. Tipičan primer je Apache Web server.

Osnovna razlika između procesa i tokova jeste da proces ima sopstveni memorijski prostor, dok tokovi dele memoriju sa svojim procesima. Mnogo je lakše rezervisati tokove i sa njima komunicirati, ali ih je teže napraviti i debugovati. Iako se u Kylixu prilično lako radi sa tokovima (i lako ih možete debugovati) i iako Linuxovi kerneli posle verzije 2.4 imaju proširene mogućnosti rada sa tokovima, njih programeri ipak retko koriste.

#### NAPOMENA

Programiranje tokova u Kylixu je veoma slično programiranju tokova u Delphiju, a jedina uočljiva razlika je toa što su prioriteti tokova brojevi a ne sistemska enumeracija. Druga razlika je što Linuxovi tokovi nemaju svojstvo `Handle`, već svojstvo `ThreadId`. Više o tokovima u poglavljima 11 i 16. ■

To je razlog zbog kojeg morate češće biti spremni na korišćenje procesa (i usmeravanja) nego u Windowsu. Videćete da je za jednostavne zadatke pokretanje procesa zaista lakše nego korišćenje toka. Dalje, za razliku od Windowsa, u Linuxu postoje funkcije koje su moćnije i lakše za korišćenje pomoću kojih se upravlja procesima. Korisnici Linuxa su već navikli da mogu "utamaniti" (engl. kill) blokirani proces, a to ne utiče na rad drugih aplikacija. Ponovno podizanje sistema u Linuxu je prilično retko jer su procesi mnogo više izolovani nego u Windowsu.

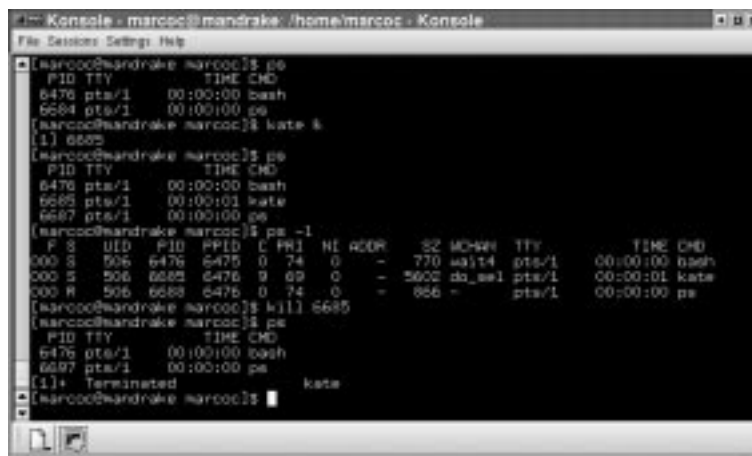
U poglavlju 16 ćete naučiti kako da koristite procese i usmeravanje u programima koje pravite pomoću Kylixa. Procese ćemo uporediti sa tokovima i razmatraćemo prednosti i nedostake ova dva rešenja. Za sada ćemo se pozabaviti komandama komandne linije koje možete koristiti za rad sa procesima.

Najjednostavniji način na koji možete napraviti proces jeste izvršavanje datoteke. U Windowsu možete pokrenuti samo EXE, COM i BAT datoteke, dok se u Linuxu svaka datoteka može pokrenuti ako ima zastavicu kojom je za tekućeg korisnika označena kao izvršna. Ako iz konzole pokrenete program, konzola će čekati da prestane izvršavanje programa kako bi ponovo mogli da je koristite. Dakle, ne možete koristiti konzolu sve dok se program izvršava. Da biste programe asinhrono pokrenuli, morate na kraj komandne linije dodati specijalni operator `&` koji školjki Bash omogućava izvršavanje komande i momentalno vraćanje kontrole konzoli.

Na slici 1.6 je prikazana lista aktivnih procesa koji su dobijeni pomoću komande `ps` i kako da ih prekinete pomoću komande `kill`. Za svaki proces komanda `ps` prikazuje Process ID (PID - identifikator procesa), jedinstveni broj kojim je identifikovan proces i terminal ili TTY sa kojeg je pokrenut. Pored toga, komanda `ps -l` vam prikazuje identifikator roditeljskog procesa (PPID). U primeru koji vidite na slici 1.6, komanda `ps` prikazuje samo procese koji su pokrenuti sa konzole. Takođe možete koristiti komandu `ps -x` kako biste prikazali sve procese korisnika ili komandu `ps -e` kako biste prikazali procese koje su pokrenuli drugi korisnici.

Iako se komanda `kill` može koristiti uz komandno ime procesa koji treba ukloniti, bezbednije je koristiti je upotrebljavajući PID procesa. Ma kako čudno izgledalo, osnovna namena komande `kill` nije prekidanje procesa, već slanje signala procesima. Signali su veoma proste poruke koje aplikacije često koriste kako bi sinhronizovale svoj rad i kako bi obavestile druge aplikacije o nekim događajima. Pošto je signal `TERM` osnovni signal kojim se zahteva da aplikacija prekine sa radom, komandu `kill` ćete često koristiti za prekidanje izvršavanja aplikacije. Zapravo, posle slanja signala, ako proces ne prihvati signal on će ipak biti prekinut.





**SLIKA 1.6** Rad sa procesima i njihovo prekidanje sa konzole

Moguće je da program prihvati signal TERM ali da se njegovo izvršavanje ne prekine zbog nekog бага ili zato što program obavlja dugačku i osjetljivu operaciju. Da biste u tom slučaju prekinuli program morate poslati specijalan signal KILL koji neće obrađivati program čime se sistem prisiljava da prekine program bez obzira u kom stanju se nalazi.

#### NAPOMENA

Signal KILL nije osnovni signal komande kill. Komandu kill-l koristite kako biste prikazali sve signale koje možete upotrebiti. Više o komandi kill pogledajte u poglavlju 16. ■

Kada Linux isključi računar, on svim procesima šalje signal TERM. Nakon toga se signal KILL šalje samo procesima koji nisu prekinuti. Ovu operaciju obično može obaviti samo administrator, ali je moguće dopustiti drugim korisnicima da zaustave sistem. Komanda koju koristite je komanda shutdown, ali pošto je morate koristiti uz navođenje parametara, ona se obično zadaje pozivanjem dva skripta, reboot i halt - koji ćete pozvati zavisi od toga da li želite da ponovo pokrenete računar ili ne.

Čak i kada nemate dozvole za zadavanje komande shutdown, računar na kome se izvršava Linux možete ponovo pokrenuti koristeći kombinaciju tastera Ctrl+Alt+Del. Ako Vam ovo izgleda nesigurno i čudno morate zapamtiti da to nije komanda konzole već tastaturna komanda što znači da se može zadati samo sa tastature koja je fizički priključena na server. Ako ste još uvek zabrinuti zbog bezbednosti, imajte na umu da svako ko može da koristi tastaturu takođe može da isključi kabl za napajanje!

Druga kombinacija tastera koja se koristi za obaranje sistema je kombinacija tastera Ctrl+Backspace pomoću koje se prekida izvršavanje XServera. Ova kombinacija tastera je naročito korisna kada neka grafička aplikacija "blokira" ili kada Vaš miš ne reaguje na akcije koje zadajete. Nasuprot Windowsu, ako neka aplikacija "zamrzne" grafički podsistem, Vi možete prekinuti rad podsistema, a da pri tom ne morate isključiti računar.

## Preusmeravanje i usmeravanje

Do sada smo sa Linuxove konzole zadavali komande jednu po jednu. Sada ćemo Vam pokazati kako da koristite operatore preusmeravanja kako biste postigli da više programa radi zajedno. Pomoću simbola `>` možete prusmeriti standardni izlaz programa u neku datoteku ili uređaj. Pomoću simbola `>>` nov izlaz možete nadovezati na kraj te datoteke. Simbol `|` Vam omogućava da standardni izlaz programa povežete sa standardnim izlazom neke druge aplikacije.

Hajde da probamo neke primere. Sledeća komanda pokreće igru nethack, skrivajući standardni izlaz (to jest, šalje ga null uređaju, vrsti crne rupe):

```
nethack > dev/null
```

Sledeća komanda dodaje red u datoteku mylog:

```
echo "I just did something" >> mylog
```

Naredne komande prikazuju pomoć za komandu tar stranicu po stranicu, broj sistemskih uređaja i sve procese koje pravi aplikacija xmms:

```
tar --help | less
ls /dev | wc
ps -e | grep xmms
```

Konačno, korišćenje komande grep uz komandu wc (word count - broj reči) pretražuje sve Kylixove direktorijume sa izvornim kodom i pronalazi i broji sva pojavljivanja reči fork:

```
grep fork -d recurse /opt/kylix/source/ | wc
```

Još jedna komanda u kojoj je korisno upotrebiti usmeravanje je sledeća komanda u kojoj se upotrebljava man2html kako bi se od stranice uputstva napravila HTML stranica:

```
man ls | man2html >ls.html
```

## Skriptovanje i promenljive okruženja

U Linuxu možete definisati promenljive okruženja. Ove promenljive podsećaju na promenljive programskog jezika, ali ih koriste svi procesi. U DOS-u takođe postoje promenljive okruženja i koriste se na sličan način (iako se sve manje i manje koriste u Windowsu i Windowsovim aplikacijama). Na slici 1.7 je prikazano kako možete definisati promenljivu i kako njenu vrednost možete prikazati u konzoli.



**SLIKA 1.7** Podešavanje promenljive okruženja i prikazivanje njene vrednosti

Tipičan način korišćenja promenljivih okruženja jeste zapisivanje putanje aplikacije. To je tipičan posao za Windowsov Registry, a u poglavlju 5 ćemo obraditi ostale slučajeve u kojima promenljive okruženja mogu efikasno zameniti Windows Registry, uključujući primer FilesList, koji pokazuje kako da čitate vrednost promenljive okruženja iz Kylix programa pomoću RTL funkcije `GetEnvironmentVariable`.

Zaljubljenici u DOS se sigurno sećaju korišćenja i pravljenja batch datoteka. U Linuxu se za automatizovanje zadataka i niza komandi školjke koriste skriptovi. U poređenju sa DOS-ovim batch datotekama, skriptovi školjke imaju manje ograničenja i poptuniji jezik; skriptovi mogu da urade gotovo sve. (To je razlog zbog kojeg se još uvek koriste, za razliku od DOS-ovih batch datoteka, koje Microsoft pokušava ponovo da uvede u upotrebu bez mnogo uspeha.) Jezik skripti školjke Bash ima toliko mnogo karakteristika da se može smatrati pravim programskim jezikom. Mi ih ne možemo sve objasniti u ovom kratkom uvodu, ali ih možete saznati ako zadate komandu `man bash`.

U DOS-u je najvažnija batch datoteka bila `autoexec.bat`, skript kojim se pokreće sistem. U Linuxu ne postoji ekvivalent jer se mnoge datoteke pozivaju za različite sistemske događaje. Najbliži ekvivalent datoteke `autoexec.bat` je skript `.bash_profile`, koji se poziva nakon svakog prijavljivanja na sistem.

Na primer, ako skriptu `.bash_profile` dodate sledeći red, konzola će Vam prikazati pozdravnu poruku:

```
echo "Welcome $USER, you are a great programmer!"
```

Za čitanje i menjanje datoteke skripta možete upotrebiti bilo koji editor teksta. Ako niste navikli na korišćenje standardnih Unixovih editora kao što su `vi`, `tex` ili `emacs`, a želite potpuni znakovni editor, možete koristiti editore `pico` ili `joe` u kojima se koriste komande prvih DOS-ovih editora kao što je `Wordstar`.

**NAPOMENA**

Programski jezik Turbo Pascal koristi komande Wordstara za kretanje kroz tekst, selektovanje i mnoge druge operacije. Neke od tih veoma starih komandi se još uvek koriste u Delphiju i Kylixu zbog kompatibilnosti. ■

Da biste izvršili skript (koji je zapravo tekstualna datoteka), navedite prvi red programa koji ga mora obraditi kojem prethode znaci `#!`. Za bash skript komanda bi bila `#!/bin/bash`, dok bi za Python skript komanda bila `#!/usr/bin/python`. Takođe, nemojte zaboraviti da, za korisnike koji će je pokretati, datoteku označite kao izvršnu.

Za Vas može biti važno da razumete Linuxove komande i skriptovanje (i jedno i drugo kao tehnike koje ćete koristiti i koje će Vas bolje uvesti u način razmišljanja u Linuxu), iako korisnik Vašeg programa verovatno očekuje nešto jednostavniji, grafički orijentisan pristup. Na sreću, da bismo odgovorili ovakvim zahtevima, mi možemo koristiti Kylix. U ovoj knjizi nećemo više govoriti o Linuxovim komandama i skriptovanju, izuzev u specifičnim slučajevima, a ako želite da se osećate kao deo sveta Linux programera i ako želite da koristite moć operativnog sistema, ne možete da zanemarite pristup koji ste upoznali u nekoliko prethodnih odeljaka. Međutim, mi ćemo se sada pozabaviti potpuno drugačijim programiranjem.

## Licenciranje softvera u Linuxu

Jedna od ključnih razlika između Windowsa i Linuxa jeste da je prvi operativni sistem vlasnički i zatvoren, dok je drugi potpuno otvoren. Ne samo da je operativni sistem besplatan, već su i mnoge Linuxove aplikacije besplatne ili postoje njihove besplatne verzije, uključujući i Kylix. Da bi razumeo ovakvu situaciju, programer mora da razume bar osnovne koncepte licenciranja u Linuxu, počevši od GPL licence.

Mi ne želimo da se upustimo u raspravu o tome koji je način licenciranja bolji za potrebe programera, industrije i krajnjeg korisnika. Postoje žučne rasprave o ovoj temi koje možete pratiti na Web sajtu slashdot.com. Umesto toga, naš cilj je da Vam damo nekoliko referenci koje se odnose na različite modele iz ugla Kylix programera.

Osnovna licenca za programiranje Linuxa je GPL licenca koju je promovisala Free Software Foundation ([www.fsf.org](http://www.fsf.org)). Softver koji možete naći pod ovom licencom se može slobodno koristiti i menjati, a i poboljšanja koja napravite se moraju slobodno prosledivati. (Ovo je kratak rezime, a ne potpuno objašnjenje. Molimo Vas da posetite Web sajt ukoliko želite da pogledate detalje.) Verzija Kylixa, Open Edition, je pod GPL licencom za softver koji pravite.

Druga licenca koja je nedavno postigla istu popularnost je MPL - Mozilla Public License ([www.mozilla.org/MPL](http://www.mozilla.org/MPL)). Možda će Vas zainteresovati činjenica da se verzija Inter Basea, open source, pojavila sa licencom ove porodice.

Za Qt, kompanija Trolltech koristi drugu licencu, iako su nedavno prihvatili GPL licencu. Prihvatanje više licenci postaje uobičajnije tako da možete odabrati paket koji odgovara Vašim potrebama. Qt licence možete naći na adresi [www.trolltech.com/developers/licensing](http://www.trolltech.com/developers/licensing), ali zapamtite da kada koristite Kylix Vi prihvataste Qt licencu koju Borland dobija od kompanije Trolltech i da se ona primenjuje na proizvod.

Na kraju, sam Kylix ima više licenci za proizvod i softver koji pravite. Pročitajte odgovarajuće datoteke koje dobijate uz proizvod. Ukratko, verzija Open Edition (i biblioteka FreeCLX koju možete dobiti na Source Forgeu) imaju GPL licencu čime se sprečava pravljenje zatvorenih aplikacija. Verzije Kylix koje plaćate Vam omogućavaju da distribuirate i prodajete programe koje napravite.

## Dobrodošli u Kylix

Mi smo Vam predstavili neke tehničke elemente operativnog sistema Linux. Sada ste spremni da se upoznate sa Kylixom. U ovom poglavlju se nećemo baviti tehničkim elementima Kylixa (na raspolaganju nam je čitava knjiga), već ćemo Vam dati samo uvod u slučaju da nikada niste koristili Kylix. Ukoliko ste ga već koristili ili ste koristili Delphi, možete preskočiti ostatak ovog poglavlja.

Kombinovanjem tehnika OOP-a (objektno-orijentisanog programiranja) i vizuelnog programiranja, Kylix spaja snagu savremenih programskih jezika i lakoću vizuelnog programiranja pomoću komponenti. Za razumevanje ovog koncepta je potrebno dosta vremena, izuzev ako već imate iskustva sa Windowsovim duplikatom Delphijem ili sličnim proizvodima (mada ih nema mnogo, čak ni u Windowsu). Za sada ću Vam samo pokazati kako da napravite prvu Kylix aplikaciju i pri tom se neću upuštati u objašnjenja kako i zašto se tako pravi. Pre nego što napravimo prvu aplikaciju, pozabaviću se instaliranjem Kylixa.

## Priprema za instaliranje

Kylixov IDE zahteva da u Linux sistemu postoje neke biblioteke. Ove biblioteke moraju biti novije verzije. Na Borlandovom sajtu možete naći spisak distribucija (koji se stalno ažurira, pa ga ja ovde neću navesti, već ću Vam savetovati da posetite stranicu [www.borland.com/kylix](http://www.borland.com/kylix)). Takođe, u Kylixovoj datoteci readme možete pročitati koje su biblioteke neophodne. Među njima su:

- glibc 2.2 ili upotpunjena verzija glibc 2.1.2
- libjpeg 6.2 ili novija biblioteka

Da biste proverili da li sistem podržava Kylix, Borland distribuira program `borpretest` koji se nalazi u direktorijumu čije je ime isto kao i ime CD-a. Pokretanjem skripta `testsystem` možete automatizovati korišćenje programa `borpretest`. Korišćenje programa je detaljno objašnjeno u tekstualnoj datoteci `preinstall`. U istoj datoteci se sugerise koje popravke su neophodne ukoliko imate stariji sistem. Ovaj test se obavlja prilikom pokretanja instalacije što vidite na slici 1.8.



**SLIKA 1.8** Poruke testiranja prilikom inicijalne faze instaliranja Kylixa

## Instaliranje Kylixa

Nakon provere sistema spremni ste da pokrenete instaliranje Kylixa (ako to već niste uradili). Instaliranje Kylixa je pojednostavljeno korišćenjem skripta i grafičkog programa za instaliranje, koji je jednostavniji od uobičajenog instaliranja Linux programa (pogledajte sliku 1.9).



**SLIKA 1.9** Kylixov grafički program za instaliranje

Najvažnija odluka koju morate doneti jeste da li softver želite da instalirate kao korisnik ili administrator. Ako ga instalirate kao korisnik, Kylix će biti instaliran u direktorijum HOME, a korisnik će moći da ga upotrebljava onako kako očekuje. Naravno, ako se na sistem prijavite kao drugi korisnik, nećete moći da koristite Kylix. Ako softver instalirate kao administrator, možete ga instalirati u deljeni sistemski direktorijum tako da ga može koristiti svaki korisnik sistema. Zbog toga ga nemojte instalirati u direktorijum /root, jer će onda moći da ga koristi samo administrator. Umesto toga, instalirajte ga u direktorijum /opt, koji se koristi za opcioni softver, ili u direktorijum /usr/local/. U oba slučaja korisnici neće moći da menjaju datoteke koje se

nalaze u Kylixovim poddirektorijumima (uključujući i primere aplikacija), izuzev ako im ne dodelite odgovarajuće dozvole.

Jednom instaliran Kylix možete pokrenuti korišćenjem datoteke `startkylix` koja podešava veliki broj promenljivih okruženja pre pokretanja Kylixa. Neke od tih promenljivih, naročito promenljive `PATH` i `LD_LIBRARY_PATH`, su neophodne za izvršavanje Kylix aplikacija koje pravite sa konzole ili menadžera datoteka.

Da biste podesili promenljive okruženja koje su neophodne za direktno izvršavanje Kylix programa (ne za korišćenje Kylixovog debagera pomoću komande `Run`), treba da koristite skript `kylixpath` tako što zadajete sledeću komandu:

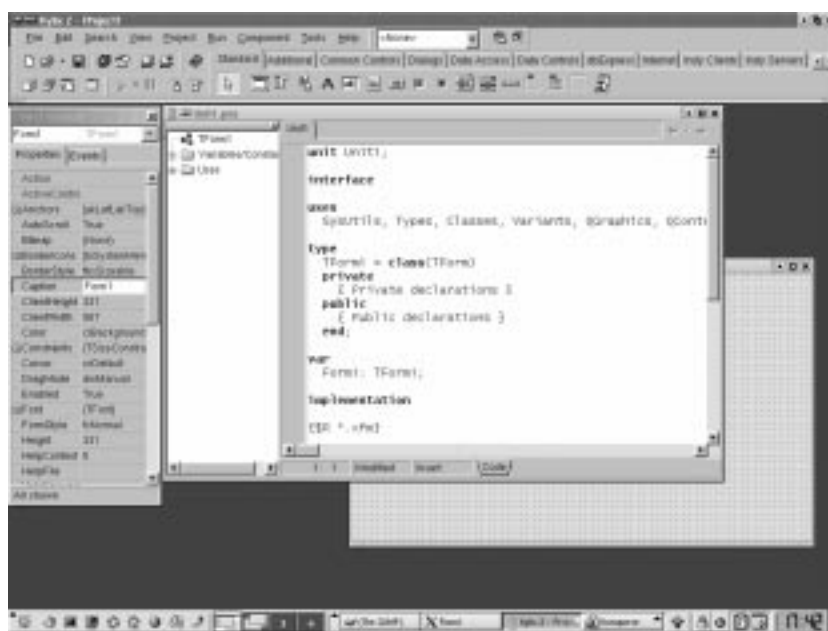
```
source kylixpath
```

Zadavanje ove komande svaki put kada otvorite konzolu je prilično zamorno pa je potrebno da u skript `.bash_profile` umetnete sledeći iskaz (kao što je opisano u Kylixovoj datoteci `readme`):

```
source <kylix directory>/bin/kylixpath > /dev/null
```

## Prvo iskustvo

Pošto ste instalirali i pokrenuli Kylix (pogledajte sliku 1.10), možete napraviti jednostavan program, počevši od konzolne verzije, a kasnije napraviti grafički program.



SLIKA 1.10 Kylixov IDE je pokrenut

## Vaša prva konzolna aplikacija

Ideja konzolne aplikacije se ponovo uvodi u Windows, ali se ovakvi programi u Microsoftovim operativnim sistemima smatraju izuzecima. U Linuxu je svaka aplikacija vezana za standardni izlaz i ulaz, uključujući aplikacije koje imaju grafički korisnički interfejs. Dakle, svaka aplikacija može zapisivati u konzolu pored toga što može koristiti grafički sistem.

Da biste nešto prikazali na ekranu, možete koristiti funkcije `write` i `writeln` koje su deo programskog jezika Pascal. Na primer, ako otvorite novu Kylix aplikaciju, njen formular uklonite pomoću komande **Project** → **Remove** menija **Project**, otvorite izvorni kod projekta koristeći komandu **Project** → **View Source** i izmenite ga u sledeći kod:

```
program Project1;

begin
    writeln ('Hello from Kylix');
end.
```

Ako ovaj program kompajlirate i pokrenete sa konzole (pošto podesite odgovarajuće Kylixove promenljive okruženja koje smo ranije razmatrali), prikazaće se tekstualna poruka.

Zapamtite da ako isti program pokrenete iz Kylixa (koristeći ugrađeni debager) nećete videti mnogo toga. Kylixov ugrađeni debager ima specifičnu opciju za otvaranje konzole i pokretanje programa iz nje. U okviru za dijalog **Run** → **Parameters** možete potvrditi polje **Use Launcher Application** kako biste videli izvršavanje programa u konzoli. Konzola će se na kratko prikazati, jer se zatvara čim se program izvrši. Potrebno je da dodate parametar (`-hold`) kako biste na ekranu zadržali izlaz nakon izvršavanja programa.

### NAPOMENA

Korišćenje konzole za pokretanje aplikacija je veoma važno u slučaju grešaka koje sprečavaju izvršavanje programa (na primer, ako nedostaje biblioteka deljenih objekata). Osnovne Linuxove biblioteke preusmeravaju poruke o greškama na uređaj `stderr` (što je obično konzola), a ne u okvir za poruku. Prednosti ove tehnike možete iskoristiti za prikazivanje Vaših poruka iz Kylixovih programa. ■

Kako ovaj program prikazuje samo poruku, odlučili smo da za propratni CD napravimo poboljšanu verziju. Aplikacija **EchoCons** ima začuđujući kod:

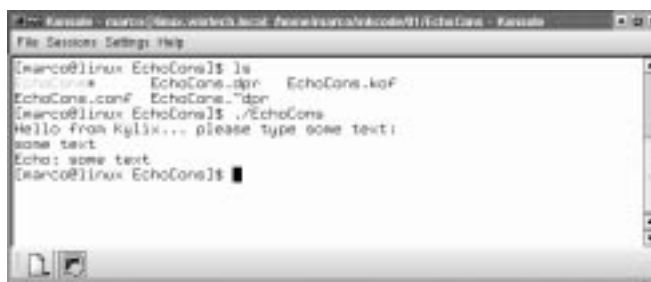
```
program EchoCons;

var
    s: string;

begin
    writeln ('Hello from Kylix... please type some text:');
    read (s);
    writeln ('Echo: ' + s);
end.
```



Kao što možete videti na slici 1.11, nakon prikazivanja poruke program čeka na Vaš unos koji zatim prikazuje na ekranu. Ovo nije složen program pa se nećemo njime baviti. Međutim, dodavanjem reda `uses libc`, možete napraviti veoma složene konzolne aplikacije u Linuxu koje mogu koristiti sve osnovne sistemske biblioteke (što nećemo raditi u ovoj knjizi, jer je to knjiga o Kylixu).



**SLIKA 1.11** Izlaz veoma jednostavne Kylixove konzolne aplikacije

### Vaša prva GUI aplikacija

Kylix se zaista može koristiti za konzolne aplikacije (uključujući specijalni podskup konzolnih aplikacija koje služe kao serverski Web programi), ali je ipak zamišljen da se koristi za pravljenje GUI aplikacija. Pravljenje slične aplikacija koja ima formular je sada još jednostavnije. Napravite novu aplikaciju, na formular smestite kontrole edit box i label, dva puta kliknite kontrolu edit box, u editoru otvorite njenu obradu događaja `OnChange` i unesite sledeće redove između ključnih reči `begin` i `end` ovog metoda:

```
procedure TForm1.Edit1Change(Sender: TObject);  
begin  
    Label1.Caption := 'Echo: ' + Edit1.Text;  
end;
```

Doradili smo ovaj program tako što smo izmenili vrednost svojstva `Caption` formulara i uneli vrednost `EchoGui` (ime programa), a sam formular smo smanjili. Ako sada pokrenete program i unesete nešto u polje u oznaci, kontrola label će prikazati tekst koji ste uneli (pogledajte sliku 1.12).



**SLIKA 1.12** Program `EchoGui`, grafička Kylixova aplikacija - nije mnogo složenija od konzolne aplikacije, ali svakako je lepšeg izgleda

## Šta je sledeće?

Svakako da smo mogli da nastavimo sa izradom ovako malih programa, ali korake koje ste preduzimali u Kylixu niste morali da razumete. Takođe, četiri reda koda koje ste videli su *metod nasleđene klase formulara koji koristi komponente koje su napravljene iz njihovih usmerenih reprezentacija pristupajući svojstvima čitanja i zapisivanja*. Ako Vam ovo nije jasno nemojte se previše brinuti: u poglavljima 3 i 4 ćete dobiti detaljna uputstva. Kada se u poglavlju 7 ponovo budemo bavili pravljenjem grafičkih aplikacija, pokazaćemo Vam mnogo naprednije primere.

U ovom poglavlju smo Vam dali samo kratak pregled tehničkih razlika između Windowsa i Linuxa, kao i veoma jednostavan primer korišćenja Kylixa. Ove osnovne informacije bi trebalo da budu dovoljne za početak, ali one svakako nisu detaljne. Svakako morate pročitati knjige o Linuxu ako želite da budete programer koji dobro poznaje ovaj operativni sistem.

Pored tehničkih detalja, prelazak na Linux može biti veliko i sjajno iskustvo za programera, naročito za one programere koji imaju dovoljno godina da se sećaju ranih dana industrije personalnih računara. Takođe, mnoge stvari koje uzimate zdravo za gotovo nisu navedene u ovom poglavlju. U mnogim slučajevima ćete morati da zaboravite i ponovo naučite neke stvari, a to može biti prilično komplikovano. No, mi smo sigurni da je sve to vredno truda i da će Vam Kylix u tome mnogo pomoći.