

Atviras kodas: Arch Linux

Maksim Norkin

2011 m. sausio 9 d.



# Turinys

<b>1</b>	<b>Įvadas</b>	<b>5</b>
1.1	Įžanga . . . . .	5
1.2	Svarbiausi Arch Linux distribucijos bruožai . . . . .	5
1.3	Licenzija . . . . .	5
1.4	Arch problemų sprendimo būdas . . . . .	6
1.5	Apie šitą vadovą . . . . .	6
<b>2</b>	<b>Pagrindinės sistemos įdiegimas</b>	<b>7</b>
2.1	Naujausios įdiegimo laikmenos gavimas . . . . .	7
2.2	Sistemos įdiegimas iš egzistuojančios GNU/Linux distribucijos . . . . .	7
2.3	Įdiegimas iš CD laikmenos . . . . .	7
2.4	Diegimas iš Flash atminties kortelės arba USB atmintinės . . . . .	7
2.5	Arch Linux įdiegimo krovimas . . . . .	8
2.6	Įdiegimo paleidimas . . . . .	9
2.6.1	Įdiegimo šaltinio pasirinkimas . . . . .	9
2.7	Laikrodžio nustatymas . . . . .	13
2.8	Kietojo disko paruošimas . . . . .	13
2.8.1	Kietojo disko skaldymas . . . . .	14
2.8.2	Direktorių pririšimas . . . . .	17
2.9	Paketų pasirinkimas . . . . .	19
2.10	Paketų įdiegimas . . . . .	19
2.11	Sistemos konfigūravimas . . . . .	20
2.11.1	/etc/rc.conf . . . . .	20
2.11.2	/etc/fstab . . . . .	24
2.11.3	/etc/mkinitcpio.conf . . . . .	25
2.11.4	/etc/modprobe.d/modprobe.conf . . . . .	26
2.11.5	/etc/resolv.conf ( statiniam IP ) . . . . .	26
2.11.6	/etc/hosts . . . . .	26
2.11.7	/etc/hosts.deny ir /etc/hosts.allow . . . . .	27
2.11.8	/etc/locale.gen . . . . .	27
2.11.9	Pacman veidrodis . . . . .	28
2.11.10	Root slaptažodis . . . . .	28
2.11.11	Pabaiga . . . . .	28
2.12	Bootloader įdiegimas . . . . .	28
2.12.1	GRUB . . . . .	28
2.13	Perkrovimas . . . . .	30

<b>3</b>	<b>Pagrindinės sistemos atnaujinimas ir konfigūravimas</b>	<b>31</b>
3.1	Žingsnis 1: Tinklo konfigūravimas	31
3.1.1	LAN tinklas	32
3.1.2	Belaidis tinklas	32
3.1.3	Proxy serveris	33
3.1.4	Analoginis modemas, ISDN ir DSL (PPPoE)	34
3.2	Žingsnis 2: Atnaujinimas su pacman	34
3.2.1	Kas yra pacman ?	34
3.2.2	Paketų saugyklos	34
3.2.3	/etc/pacman.conf	36
3.2.4	/etc/pacman.d/mirrorlist	36
3.2.5	Mirrorcheck sistemos naujumui tikrinimui	37
3.2.6	Paketų atnaujinimo ignoravimas	38
3.2.7	Konfigūracinių bylų ignoravimas	38
3.2.8	Supažindinimas su pacman	38
3.3	Sistemos atnaujinimas	38
3.3.1	Arch dabartinės versijos modelis	39
<b>4</b>	<b>Naujo vartotojo sukūrimas</b>	<b>41</b>
4.0.2	Vartotojo pašalinimas	42
4.1	Sudo diegimas ir konfigūravimas ( nebūtina )	43
<b>5</b>	<b>Garsas</b>	<b>45</b>
5.0.1	Garso testavimas	46
5.0.2	Garso nustatymų išsaugojimas	46
<b>6</b>	<b>Grafinė vartotojo aplinka</b>	<b>47</b>
6.1	Žingsnis 1: X diegimas	47
6.1.1	A: Xorg diegimas	47
6.1.2	B: Vaizdo tvarkyklės diegimas	47
6.1.3	C: Įvesties ( input ) tvarkyklių diegimas	50
6.2	Žingsnis 2: X konfigūravimas ( nebūtina )	51
6.2.1	Ne-US klaviatūra	51
6.3	Žingsnis 3: Bazinės grafinės aplinkos paleidimas	51
6.3.1	Pranešimų magistralė	52
6.3.2	X paleidimas	52
6.3.3	Klaidų atveju	53
6.3.4	Reikia pagalbos?	53
6.4	Žingsnis 4: Šriftų diegimas	53
6.5	Žingsnis 5: Grafinės aplinkos pasirinkimas ir diegimas	54
6.5.1	Darbastalio aplinka	54
6.5.2	Langų tvarkyklės	54
6.6	Grafinės aplinkos paleidimo metodai	54
6.6.1	A: Rankinis	54
6.6.2	B: Automatinis	55

# skyrius 1

## Įvadas

### 1.1 Įžanga

Sveikas skaitytojau.

Šita knyga padės Jums susipažinti, įdiegti ir susikonfigūruoti Arch Linux sistema Jūsų kompiuteryje. Arch Linux yra lengva, mažai vietos užimanti GNU/Linux distribucija, orientuota į kompetetingus vartotojus. Šitas vadovas yra orientuotas į naujus Arch vartotojus, tačiau tinka ir patyrusiam vartotojui, kaip informacijos šaltinis.

### 1.2 Svarbiausi Arch Linux distribucijos bruožai

- Paprasta filosofija ir paprastas dizainas;
- Visi paketai kompiliuojami i686 ir x86\_64 architektūroms;
- BSD stiliaus paleidimo skriptai, kurie valdomi iš vieno, centrinio failo;
- mkinitcpio: Paprastas ir dinamiškas initramfs kūrėjas;
- Pacman paketų tvarkyklė yra lengva ir lanksti, reikalaujanti mažų sistemos resursų;
- Arch Build System yra paketų kūrimo sistema, paremta portų architektūra, kuri suteikia paprastą kiautą kuriant įdiegiamus Arch paketus iš pradinio kodo;
- Arch User Repository siūlo daugiau nei tūkstantį vartotojų sukurtų kūrimo skriptų ir galimybę patalpinti savo sukurtą kūrimo skriptą.

### 1.3 Licenzija

Arch Linux, pacman, dokumentacijos ir skriptų autorinės teisės 2002-2007 metais priklauso Judd Vinet, o nuo 2007 priklauso Aaron Griffin. Autorinės teisės saugo GNU General Public Licenzijos 2 versija.

## 1.4 Arch problemų sprendimo būdas

Visos Arch distribucijos architektūra yra paremta vienu principu - viską išlaikyti kuo paprastesniam variante.

Paprastumas šiame kontekste reiškia - be jokių bereikalingų pridėjimų, modifikacijų ar kompiliavimų. O trumpai - elegantiškas ir minimalistinis problemos sprendimo būdas.

Kelios mintys apie paprastumo supratimą:

- “Techniniu požiūriu ‘paprastumas’ nėra stabili pozicija. Geriau būti techniškai elegantišku su pritaikomomis žiniomis, negu būti lengvu naudojime ir technišku.” - Aaron Griffin
- “Subjektai neturi būti manipuluojami, jeigu tai nėra būtina” - Occam skustuvai. Skustuvai šiuo atveju vaidina nereikalingų operacijų pašalinimą, leidžiantį tęsti priėjimą prie paprasto paaiškinimo, metodo ar teorijos.
- “Neįprasta mano technikos dalis remiasi paprastumu.. Progreso dydis visą laiką remiasi paprastumo principu.” - Bruce Lee

## 1.5 Apie šitą vadovą

Arch wiki yra labai geras informacijos šaltinis, tad pirmiausiai, kreipdamiesi pagalbos, įsitikinkite, kad Jūsų problema nėra aprašyta wiki puslapyje. Jeigu atsakimo į savo problemą taip ir neradote, galite kreiptis į #archlinux IRC kanalą freenode serveryje arba galite kreiptis į forumą - <http://bbs.archlinux.org>; Taip pat galite kreiptis į Arch Linux Lietuva bendruomenę <http://sls.archlinux.lt>.

Visas vadovas yra suskirstytas į penkias dalis:

- Dalis 1: Pagrindinės sistemos įdiegimas;
- Dalis 2: Arch Linux pagrindinės sistemos atnaujinimas;
- Dalis 3: Naujo vartotojo sukūrimas
- Dalis 4: Garsas;
- Dalis 5: Grafinė vartotojo aplinka;
- Išvados

## skyrius 2

# Pagrindinės sistemos įdiegimas

### 2.1 Naujausios įdiegimo laikmenos gavimas

Arch Linux oficialią įdiegimo laikmeną galite gauti iš <http://archlinux.org/download>. Vadovo rašymo metu, naujausia versija yra 2010.05.

- Tiek *Core* tiek *Netinstall* atvaizdai suteikia tik pagrindinę sistemą. Verta pastebėti, jog pagrindinėje Arch Linux sistemoje nėra jokios grafinės aplinkos. Pagrindinė sistema susideda iš GNU įrankių grandinės (kompiliatoriaus, asmeblerio, linkerio, kt. ), Linux branduolio ir kelių papildomų bibliotekų ir modulių.
- Įdiegimas yra palengvintas tiek *Core*, tiek *Netinstall* atvaizduose.
- *Netinstall* atvaizdas yra mažesnis, bet jame visiškai nėra pagrindinės sistemos paketų. Visa sistema yra parsiumčiama iš interneto.

### 2.2 Sistemos įdiegimas iš egzistuojančios GNU/Linux distribucijos

Arch Linux yra pakankamai lankstus, kad galėtų būti įdiegtas iš kitos, egzistuojančios distribucijos į laisvą patriciją arba iš Live CD. Įdiegimą iš egzistuojančios GNU/Linux distribucijos apžvelgsime vadovo pabaigoje.

### 2.3 Įdiegimas iš CD laikmenos

Iškepkite atsiųstą .iso atvaizdą į CD arba DVD su mėgstama CD/DVD rašymo programa ir tęskite savo kelią į kitą skyrių *Arch Linux įdiegimo krovimas*

### 2.4 Diegimas iš Flash atminties kortelės arba USB atmintinės

Sekantis metodas veiks bet kokiam Flash atminties tipui, kurį BIOS leis krauti paleidimo metu, būtų tai kortelių skaitytuvas arba USB portas.

**UNIX Metodus** Įdėkite tuščią arba nereikalingą flash laikmeną, nustatykite iki jos kelią ir įrašykite .iso atvaizdą, pasitelkus `/bin/dd` programą:

```
dd if=archlinux-2010.05-{core|netinstall}-{i686|x86_64|dual}.iso\
of=/dev/sdx
```

kur *if* = yra kelias iki atvaizdo failo ir *of* = yra kelias iki Jūsų flash laikmenos. Įsitikinkite, kad naudojate `/dev/sdx`, o ne `/dev/sdx1`. Jums reikės flash atminties tiek, kad joje tilptų 381MB duomenų.

**Patikrinkite md5sum** Pasižymėkite koks buvo įrašų (blokų) skaičius, kai buvo rašoma į laikmeną. Tuomet galima patikrinti:

```
dd if=/dev/sdx count=irasu_skaicius status=noxfer | md5sum
```

Patikrinimo kodas turi sutapti su parsųsto atvaizdo md5sum.

## 2.5 Arch Linux įdiegimo krovimas

Įdėkite CD ar Flash laikmeną, perkraukit kompiuterį ir paleiskite sistema iš CD ar Flash laikmenos. Jums gali prireikti pakeisti krovimosi eiliškumą BIOS nustatymuose arba paspausti kažkokį mygtuką. Dažniausiai toks mygtukas būna DEL, F1, F2, F11 arba F12. Pabandykite paspausti vieną iš jų, kuomet BIOS yra POST ( Power On Self-Test ) režime.

Pastaba: Atminties reikalavimai baziniam įdiegimui yra tokie:

- *Core* : 128 MB RAM x86\_64/i686 ( pažymėti visi paketai, su swap partcija )
- *Netinstall* : 128 MB RAM x86\_64/i686 ( pažymėti visi paketai, su swap partcija )

Šiame žingsnyje turėtų pasirodyti pagrindinis menu. Pasirinkite Jums reikalingą opciją klaviatūros navigaciniais mygtukais ir padarę pasirinkimą paspauskite 'Enter' mygtuką.

Dažniausiai, pirmą kartą kraunant Arch Linux *Boot Archlive* yra tas pasirinkimas, kurio Jums reikia. Tačiau, jeigu turite bėdų su libata/PATA arba neturite SATA (Serial ATA), pasirinkite *Boot Archlive [legacy IDE]*.

Norint pakeisti GRUB pasirinkimus, paspauskite raidę **e**. Dauguma vartotojų norės pakeisti framebuffer rezoliuciją, patogesniai darbui. Pridėkite:

```
vga=773
```

kernelio eilutėje. Tuomet paspauskite **Enter** mygtuką, kad įgalinti 1024x768 framebuffer rezoliuciją. Kuomet viskas bus padaryta, paspauskite **b** mygtuką, kad pradėti sistemos krovimą.

Dabar sistema turi pradėti krautis. Kuomet sistema pilnai pasikraus, turėtų pasirodyti prisijungimo galimybė. Prisijunkite kaip *root* vartotojas, kadangi Archlive diske iš *root* vartotojo nereikalaujama slaptažodžio.



**Klaviatūros išdėstymo keitimas** Jeigu turite ne US klaviatūros išdėstymą, tuomet Jums reikia ją pakeisti. Tai galima atlikti su *km* komanda:

```
km
```

arba panaudoti *loadkeys* komandą:

```
loadkeys layout
```

( pakeiskite *layout* su Jūsų pasirinktu klaviatūros išdėstymu ).

Dauguma lietuviškų klaviatūrų yra US standarto su papildomais lietuviškais simboliais, kurie pasirodo vietoj skaičių eilės, esančios virš *q,w,e,r,t,y,u,i,o,p* klavišų eilės. Tai reiškia, kad daugumoje atveju, Jums visiškai nereikia keisti klaviatūros išdėstymo.

**Dokumentacija** Oficialus įdiegimo vadovas yra pasiekiamas iškarto diske. Norint jį pasiekti, persijunkite į kitą konsolę ( *ALT+F2* ) ir tuomet surinkite tokią komandą:

```
less /usr/share/aif/docs/official_installation_guide_en
```

*Less* komanda leis Jums peržvelgti visą vadovą puslapiais. Norint persijungti atgal į įdiegimą, tiesiog paspauskite *Alt+F1* ir grįšite į pirmą konsolę, kurioje yra vykdomas įdiegimas.

Prireikus vėl paskaityti dokumentaciją, tiesiog persijunkite į antrą terminalą su *Alt+F2*, norint grįžti prie įdiegimo - *Alt+F1*.

Pastaba: Įsidėmėkite, kad oficialus įdiegimo vadovas apžvelgia tik pagrindinės sistemos įdiegimą ir konfigūravimą. Kai tik su pagrindine sistema yra susitvarkyta, rekomenduojama grįžti prie detalesnio vadovo, kuriame yra aprašyti visi reikalingi žingsniai po įdiegimo.

## 2.6 Įdiegimo paleidimas

Kuomet esate prisijungę kaip *root* vartotojas, pirmame terminale galite paleisti įdiegimo skriptą:

```
/arch/setup
```

### 2.6.1 Įdiegimo šaltinio pasirinkimas

Po pasisveikinimo, Jūsų paprašys pasirinkti įdiegimo šaltinį. Priklausomai nuo ankstesnės laikmenos pasirinkimo, atitinkamai pasirinkite ir įdiegimo šaltinį.

- Jeigu pasirinkote *Core* įdiegimo atvaizdą, tęskite toliau prie skyriaus *Laikrodžio nustatymas*
- Jeigu pasirinkote *Netinstall*, Jums reiks rankiniu būdu užkrauti tinklo plokštės tvarkykles ( žinoma, jeigu sistema automatiškai neaptiks Jūsų turimos įrangos ). Udev yra labai naudingas įrankis, norint sužinoti Jūsų turimą įrangą. Tai galima patikrinti, pasitelkus *ifconfig -a* komandą.

**Tinklo konfigūravimas (Netinstall)** Šitam žingsnyje, sistema turi parodyti jos rastus tinklo sąsajas. Jeigu sąsaja ir HWaddr ( HardWare address ) yra sąrašė, tuomet Jūsų tinklo plokštė buvo sėkmingai aptikta ir jos tvarkyklės sėkmingai įkrautos į branduolį. Jeigu Jūsų tinklo plokštė atpažinta nebuvo, tuomet Jums reikia rankiniu būdu kelti tvarkykles į branduolį kitoje konsolėje.

Atsiradus panašiam langui, bus paprašyta pasirinkti *sąsają*, *zoną* arba *Atšaukti*. Pasirinkite reikiamą *sąsają* ir tęskite.

Paskui įdiegimo programa paklaus Jūsų ar Jūs norite naudoti *DHCP*. Pasirinkus *Yes*, įdiegimo programa paleis **dhcpcd**, kuri aptiks tinklo sąsaja ir iš sąsajos reikalaus IP adreso. Jeigu pasirinksite *No*, tuomet įdiegimo programa paprašys Jūsų įvesti statišką IP, tinklo kaukę, transliatorių, tinklo sąsajos DNS IP, HTTP ir FTP proxy. Paskutiniame žingsnyje sistema parodys gautus nustatymus. Šitame žingsnyje galite patikrinti ar visi nustatymai yra teisingi.

**Greitasis (A)DSL paleidimas** ( Jeigu turite modemą arba routerį tilto režime, jungiantis per ISP jungtį )

Persijunkite į kitą konsolę ( ALT+F2 ), prisijunkite kaip *root* vartotojas ir rašykite:

```
pppoe-setup
```

Jeigu viskas sukonfigūruota teisingai, pabaigoje galite prisijungti prie savo ISP su:

```
pppoe-start
```

Grįžkite į pirmą konsolę ( ALT+F1 ) ir tęskite įdiegimą *Laikrodžio nustatymas* skyriuje.

**Greitasis bevielio tinklo paleidimas** ( Jeigu turite bevielio ryšio galimybę įdiegimo metu )

Šiuo metu bevielio ryšio tvarkyklės yra pasiekiamos ir iš įdiegimo aplinkos atvaizdo. Geras savo bevielio įrangos žinojimas šiuo atveju Jums labai padės. Įsidėmėkite, jog sekantys nustatymai galios tik dabartinėje aplinkoje. Kuomet baigsite įdiegimą, visus nustatymus reiks pakartoti įdiegtoje sistemoje, norint ir toliau naudotis bevieliu ryšiu.

Taip pat verta pastebėti, jog šie žingsniai nėra privalomi, jeigu įdiegimas yra įmanomas ir be bevielio tinklo. Bevielio tinklo konfigūravimas gali būti atliktas ir po įdiegimo pabaigos, jau įdiegtoje sistemoje.

Pagrindiniai žingsniai būtų tokie:

- Persijunkite į laisvą konsolę ( pavyzdžiui 3, ALT+F3 );
- Prisijunkite kaip *root* vartotojas;
- ( Nebūtina ) Identifikuokite savo bevielio ryšio sąsają:

```
lspci | grep -i net
```

- Įsitikinkite, kad *udev* atpažino bevielio ryšio kortą ir pakrovė į branduolį reikalingus modulius. Tai galima patikrinti su */usr/bin/iwconfig* komanda:

```
iwconfig
```

Rezultatas turėtų būti kažkas panašaus:

```
lo no wireless extensions.
eth0 no wireless extensions.
wlan0    unassociated  ESSID:""
        Mode:Managed Channel=0 Access Point: Not-Associated
        Bit Rate:0 kb/s Tx-Power=20 dBm Sensitivity=8/0
        Retry limit:7 RTS thr:off Fragment thr:off
        Power Management:off
        Link Quality:0 Signal level:0 Noise level:0
        Rx invalid nwid:0 Rx invalid crypt:0 Rx invalid frag:0
        Tx excessive retries:0 Invalid misc:0 Missed beacon:0
```

Šiame pavyzdyje, *wlan0* ir bus bevielio ryšio sąsają, per kurią galima jungtis prie bevielio tinklo.

- Sekantis žingsnis būtų įgalinti sąsają su */sbin/ifconfig <sąsaja> up* komanda. Pavyzdys, turint *wlan0* sąsają:

```
ifconfig wlan0 up
```

Įsidėmėkite, jog Jūsų sąsajos pavadinimas gali būti kitoks, negu *wlan0*. Priklausomai nuo modulio, kurį užkrovė *udev*, Jūsų sąsajos pavadinimas gali būti *wlan0*, *eth1*, kt.

- Jeigu nežinote tiksliai savo bevielio tinklo *ssid* arba norite prisijungti prie Jums nežinomo bevielio tinklo, naudokite */sbin/iwlist <sąsaja> scan* komanda.

```
iwlist wlan0 scan
```

Ši komanda peržvelgs šiuo metu pasiekiamus bevielio ryšio tinklus.

- Sekantis žingsnis būtų nurodyti Jūsų bevielio tinklo sąsajai tinklo *ssid*. Priklausomai nuo bevielio tinklo kodavimo algoritmo (WEP, WPA arba išvis jokio), nurodymo procedūra gali skirtis. Pavyzdžiui galima padaryti prielaidą, jog Jūs norite prisijungti prie *atviras-tinklas*;

- Jeigu tinklas nekoduojamas:

```
iwconfig wlan0 essid "atviras-tinklas"
```

- Jeigu tinkle naudojamas WEP kodavimas su aštuntainiu apsaugos kodu:

```
iwconfig wlan0 essid "atviras-tinklas" key 25dfb2575a
```

- Jeigu tinkle naudojamas WEP kodavimas su ASCII slaptažodžiu:

```
iwconfig wlan0 essid "atviras-tinklas" key s:slaptaszodis
```

- Tinkle naudojant WPA, prisijungimo procedūra šiek tiek tampa sudėtingesne. Vėliau tokius pat rezultatus galima pasiekti pasitelkiant *netcfg* įrankį. Tolesnės komandos sugeneruos Jums konfigūracinius failus ir pabandys prisijungti prie tinklo:

```
wpa_passphrase atviras-tinklas "slaptazodis" >>/etc/wpa_supplicant.conf
wpa_supplicant -B -Dwext -i wlan0 -c /etc/wpa_supplicant.conf
```

Verta pastebėti, jog antra komandai reikia žinoti kokias tvarkykles naudoja bevielio tinklo sąsaja. Šiuo atveju tai *wext*. Prieš rašant šią komandą verta patikrinti kokias tvarkykles į branduolį įkrovė *udev*.

- Patikrinti ar visas procesas įvyko sėkmingai galima su:

```
iwconfig wlan0
```

- Toliau sekantis žingsnis būtų paprašyti bevielio tinklo prisijungti su */sbin/dhpcd* <sąsaja>. Pavyzdžiui:

```
dhpcd wlan0
```

Norint pamatyti kreipimosi detales, galima pridėti *-d* opciją ir *dhpcd* parodys visus savo žingsnius:

```
dhpcd -d wlan0
```

- Paskutinis žingsnis būtų patikrinti ar *dhpcd* sėkmingai sujungė su bevielio tinklo maršrutizatoriumi ir Jūs turit internetą:

```
ping -c 3 www.google.lt
```

Dabar Jūs jau turėtumėt būti prisijungę prie interneto ir dabar galite sėkmingai tęsti Arch Linux įdiegimą į Jūsų kompiuterį, naudojantis *Netinstall* atvaizdą.

**Ar mano bevielio tinklo korta reikalauja Firmware?** Labai mažas bevielio tinklo kortų, kartu su tvarkyklėmis, reikalauja ir firmware. Tai galima patikrinti su */usr/bin/dmesg* komanda.

```
dmesg | grep firmware
```

Kaip pavyzdį galima paimti Intel kortas, kurios reikalauja naudoti kartu su tvarkyklėmis ir firmwarą:

```
firmware: requesting iwlwifi-5000-1.ucode
```

Jeigu *dmesg* komanda nieko neparodė, vadinasi, Jūsų korta nereikalauja jokio firmware.

## 2.7 Laikrodžio nustatymas

Konfigūruojat sistemą įdiegimo metu, Jūsų bus paprašyta pasirinkti laikrodžio konfigūracijos tipą. Yra du laikrodžio konfigūracijos tipai: UTC ir localtime.

- Pasirinkite UTC, jeigu Jūsų kompiuteryje veikia tik *unix* tipo operacinės sistemos.
- Pasirinkite localtime, jeigu Jūsų kompiuteryje veikia ir kito tipo operacinės sistemos ( pavyzdžiui Windows 7 ).

## 2.8 Kietojo disko paruošimas

**Svarbu** Kietojo disko skaldymo procesas gali negrįžtamai pašalinti visus duomenys, esančius diske. Tęsiant sekančius žingsnius, patariama padaryti atsarginę visų Jums svarbių duomenų kopiją kitoje laikmenoje.

**Svarbu** Sekančiuose žingsniuose pasirinkus “Cancel” kietojo disko paruošimas nenutrūks. Norint nutraukti kietojo disko paruošimo procedūrą, reikia visiškai išeiti iš įdiegimo. Tai galima padaryti *Ctrl+C* komanda. Tik tokiu atveju visi procesai bus sustabdyti ir Jūsų kietasis diskas liks nepakitęs.

**Pastaba** Visiškai nebūtina ruošti diską Arch Linux įdiegimui iš Arch Linux įdiegimo atvaizdo. Tai galima atlikti ir iš kitos distribucijos ar net Windows operacinės sistemos, pasitelkus disko paruošimo įrankius, tokius kaip *Gparted* ir kt. Jeigu diskas buvo suskaldytas ir paruoštas anksčiau, galite tęsti *Failų sistemos pajungimas* temoje.

Patikrinkite, jog sistema atpažino Jūsų diską ir jo architektūrą, pasitelkiant */sbin/fdisk* komanda su *-l* opcija.

Persijunkite į kitą konsolę ( *Alt+F3* ) ir įveskite

```
fdisk -l
```

Visai gerai būtų įsiminti kokius disko skirsnius Jūs norite skirti Arch Linux sistemai. Dar geriau būtų - užsirašyti.

Persijunkite atgal į Arch Linux įdiegimą ( *Alt+F1* )

Pasirinkite patį pirmą menu įrašą “Prepare Hard Drive”.

- 1 pasirinkimas: “Auto Prepare”.  
“Auto Prepare” suskirsto Jūsų diską tokia architektūra:
  - */boot* skirstinys su *ext2* failų sistema. Numatytas dydis yra 32MB. Sistema gali suteikti Jums galimybę pakeisti */boot* skirstinio dydį. Jeigu planuojate turėti daug branduolių, sveika pasirinkti šiek tiek daugiau ir dar plius kažkiek atsargai ( Tarkim 500MB ).
  - *swap* skirstinys. Numatytas skirstinio dydis yra 256MB. Sistema gali suteikti Jums galimybę pakeisti *swap* skirstinio dydį. Rekomenduojamas *swap* skirstinio dydis: *turimas Ram atminties kiekis x 2*

- Atskiri `/` ir `/home` skirstiniai, kurių dydžius taip pat galima keisti. Galima pasirinkti iš sekančių failų sistemos tipų: *ext2*, *ext3*, *ext4*, *reiserfs*, *xfs* ir *jfs*. Rekomenduojama pasirinkti tokias pat failų sistemas tiek `/`, tiek `/home` skirstiniams.

Verta pastebėti, jog “Auto Prepare” visiškai pertvarkys turimą diską, t.y. ištrins visą informaciją ir visiškai perrašys disko architektūrą.

**Pastaba** “Auto Prepare” siūloma pasirinkti tiems, kurie pirmą kartą įdieginėja Arch Linux sistemą, pirmą kartą mato `/sbin/fdisk` ir labai bijo. Vėliau, šiek tiek įgudus, sistemą vėlgi galima bus iš naujo įdiegti, tik šiuo atveju protingai suskirstyti diskų skirstinius.

- 2 pasirinkimas: “Partition Hard Drives” ( su *fdisk* )

Šitas pasirinkimas leis keisti diskų architektūrą ir skirstinių tipus taip, kaip nori vartotojas.

Tie vartotojai, kurie jau yra pažįstami su *fdisk* ir disko skirstinių sudarymų, gali praleisti sekančią temą ir tęsti *Paketų pasirinkimas* temoje, kuri yra toliau.

## 2.8.1 Kietojo disko skaldymas

**Skirstinio informacija** Kietojo disko skaldymas sudaro diske tam tikras atminties sritis, kurios vadinamos skirstiniais, kurie elgiasi kaip savarankiški diskai.

Yra trijų tipų skirstiniai:

- Primary
- Extended
- Logical

*Primary* skirstiniai gali būti kraunamieji skirstiniai ( *bootable* ), tačiau jų skaičius yra limituotas iki 4 viename kietame diske. Jeigu sistemoje reikalaujama turėti daugiau kaip 4 skirstinius, tuomet reikia naudoti kitą skirstinio tipą - *extended*, kuris susidės iš *logical* skirstinių.

Pats *extended* skirstinys yra beveiksmis ir bevertis, kadangi pats *extended* skirstinys yra kaip talpa kitiems *logical* skirstiniams.

**Swap skirstinys** Swap skirstinys yra rezervuota kietojo disko talpa, kurią sisteminis branduolys galės naudoti kaip ‘RAM’ atmintį, jeigu realios ‘RAM’ atminties neužteks duotai užduočiai įvykdyti.

Istoriškai, pagrindinis swap skirstinio reikalavimas yra *x2* realios RAM atminties dydis. Bėgant laikui, šita taisyklė tampa neaktuali, kadangi pagrindinės atminties dydis asmeniniuose kompiuteriuose labai padidėjo. Šiuo metu, jeigu Jūsų kompiuteryje yra *512MB* pagrindinės atminties - dar galima taikyti *x2* taisyklę, tačiau, jeigu Jūsų kompiuteryje yra *1024MB* atminties - apie swap skirstinį galite net pamiršti. Kodėl? Kadangi visuomet lieka galimybė susikurti ne swap skirstinį, o swap failą, kuris atliktų tą patį vaidmenį, ką atlieka ir swap skirstinys.

Šiame vadove mes naudosime *1GB* swap skirstinį.

**Pastaba** Jeigu planuojate naudoti suspend-to-disk (hibernate) swap skirstinys turi būti mažų mažiausiai lygus RAM atminties dydžiui. Kai kurie Arch vartotojai rekomenduoja swap skirstinį padaryti 10-20 proc didesnį, palikti atsargą klaidos atvejui.

**Skirstinių schema** Skirstinių schema yra labai asmeninis reikalas. Nėra vieningo, numatyto geriausiu sprendimo skirstinių schemai. Kiekvieno vartotojo skirstinių schema bus skirtingai pritaikyta pagal jo poreikius ir darbo sistemos reikalavimus.

Skirstinius galima “pririšti” prie tam tikrų direktorių. Galimos direktorijos ir jų apibūdinimas:

- **/ ( root )** Root failų sistema yra pagrindinė failų sistema, iš kurios seka kitos failų sistemos. Jinai yra failų sistemos hierarchijos viršūnėje. Kiekviena direktorija seka po /, netgi tuomet, kai kažkuri direktorija yra pririšta prie kito disko.
- **/boot** Direktorija saugoja sisteminių branduolį, ramdisk atvaizdus ir krovėjo ( grub ) konfigūraciją. /boot saugo visą informaciją, kuri reikalinga branduoliui, prieš paleidžiant sistemą. Tokia informacija gali būti išsaugota *master boot sectors* arba *sector map* failai. /boot yra naudojamas tik sistemos krovimuisi, tačiau /boot gali būti iškeltas ir į kitą disko skirstinio vietą.
- **/home** Suteikia pakatalogį, kurio kiekvienas vardas sutampa su sistemoje priregistruvaisiais vartotojų vardais. /home saugoma visa informacija, susijusi su vartotojo veikla, vartotojo asmeniniais duomenimis ir vartotojo programų nustatymais.
- **/usr** Šita direktorija pagal svarbą eina iškart po / ( root ) direktorijos. Joje yra saugoma tokia informacija, kuri pasiekama visiems sistemoms vartotojams, įskaitant ir programas bei tvarkykles, kurias gali naudoti iškart keli vartotojai. /usr yra sistemiskai prieinama, tačiau ją galima tik skaityti. Tai leidžia suteikti prieigą tinklame esančiam vartotojui pasiekti /usr direktoriją, tačiau vartotojas negali tos informacijos perrašyti ar kitaip pakeisti.
- **/tmp** Direktorija atlieka “laikinos failų saugyklos” vaidmenį. Kokia informacija gali būti laikina? Tai gali būti programų *užraktai* - '.lck' tipo failai, kurie yra naudojami sudarant sąlygas neleisti tai pačiai programai paleisti savo kopijas iš skirstinių vartotojų vienu metu. Visi duomenys iš /tmp yra pašalinami po kiekvieno sistemos užkrovimo.
- **/var** Direktorija saugo savyje visą informaciją apie kintamuosius: *spool* direktorijas ir failus, administravimo ir žurnalo ( log ) failus, *pacman* puodynę, ABS medį ir kt. Taip pat /var atlieka svarbų vaidmenį - suteikia galimybę užkrauti /usr skaitymo režime. Viskas, kas istoriškai buvo talpinama /usr direktorijoje, priklausė nuo /var direktorijos.

Kiekvienam vartotojui gali iškilti klausimas - o kam man skaldyti diską į skirstinius ir paskui “pririšinėti” direktorijas prie atskirų skirstinių? Juk realiai galima padaryti du skirstinius - vienas swap, kitas / ( root ) direktorijai ir viskas.

Taip, ir toks variantas yra įmanomas, tačiau, disko skaldymas ir skirstinių pririšimas prie direktorijų turi savo privalumų:

- Saugumas: Kiekviena direktorija */etc/fstab* faile gali būti sukonfigūruota būti 'nosuid', 'nodev', 'noexec', 'readonly' ir kt.
- Stabilumas: Vartotojas, arba kvaila programa gali visiškai užteršti diską bereikalingą informaciją, arba kritiškai reikalinga informacija ( ypač */boot* skirstinyje ) gali būti pažeista ir sistema daugiau pati neužsikraus.
- Greitis: Failų sistema, į kurią yra rašoma labai dažnai, gali patapti fragmentiška ( Geriausias būdas išvengti tokios problemos, yra nuolat stebėti ar failų sistema neužima viso disko talpos ). Atskiros failų sistemos yra atskirtos ir tarpusavyje negali sąveikauti.
- Vientisumas: Jeigu viena failų sistema sugadinama - kitos lieka nepažeistos.
- Universalumas: Duomenų dalinimasis tarp kompiuterių tampa daug saugesnis, kuomet naudojami atskiri diskai duomenims perduoti.

Šiame vadove, mes atskirsim */*, */var*, */home* ir *swap* skirstinius.

**Pastaba** */var* direktorija susideda iš daug ir mažų failų. Į tai reikia atsižvelgti, renkant failų sistemos tipą.

**Kokie turi būti mano skirstinių dydžiai?** Toks klausimas yra atsakomas kiekvieno atskirai. Paprasčiausiais atvejais galima apsiriboti tik root ir swap skirstiniais, arba dar paprasčiau - vien tik root skirstiniu, be swap. Taip pat galite paeksperimentuoti ir pasirinkti tokią pat skirstinių ir direktorijų architektūrą, kaip ir mūsų pavyzdyje:

- Root failų sistema ( */* ) turės savyje */usr* direktoriją, kuri yra gan didelė. Į tai turi būti atsižvelgta. Apie 15-20GB šiai direktorijai turėtų būti užtektinai kiekvienam vartotojui.
- */var* failų sistema susidės iš daugybės duomenų, tarp kurių ir ABS medis, bei pacman puodynė. Paketų saugojimas kompiuteryje turi savo privalumų - bet kada galima įdiegti senesnę programos versiją. */var* turi tendencija didėti - pacman puodynė didėja nuolatos ir jina gali labai didelę po ilgo pacman naudojimo. Jeigu naudojate SSD tipo diską, patartina perkelti */var* direktoriją į HDD tipo diską, taip išsaugant SSD tipo diską nuo bereikalingų rašymo/skaitymo operacijų. Darbiniam kompiuteryje */var* direktorijai galima skirti 8-12GB disko vietos. Serveriai pasižymi ypač dideliais */var* direktorijos matmenimis.
- */home* direktorija yra ta vieta, kur saugomi vartotojo asmeniniai duomenys, atsiųstas turinys ir kt. Darbiniame kompiuteryje, */home* direktorija dažniausiai yra pati didžiausia. Verta prisiminti, jog jeigu Jūs iš naujo diegsite Arch Linux - Jums */home* direktorija niekur nedings ( jeigu "pririšote" */home* direktoriją prie kito skirstinio ).



- Kiekvieno disko dydžio 25 procentai nueina apsaugojimui nuo neprognozuotų atsitikimų, sistemos išplėtimo ir vaidins gerą priemonę prieš fragmentaciją.

**Skirstinių kūrimas su cfdisk** Pradėkite sukurdami naują skirstinį, kuris bus pririštas prie root failų sistemos.

Pasirinkite **New** -> **Primary** ir įveskite norimą root failų sistemos skirstinio dydį. Padėkite diską schemos pradžioje.

Taip pat pasirinkite **Type '83 Linux'**. Naujai sukurtas skirstinys, kuris bus pririštas prie / ( root ) failų direktorijos, turėtų pasirodyti *sda1* pavadinimu.

Dabar sukurkite dar vieną *primary* '83 Linux' tipo skirstinį. Šitas skirstinys bus pririštas prie /var failų direktorijos. Jis turėtų pasirodyti *sda2* pavadinimu.

Sekantis žingsnis būtų sukurti *swap* skirstinį. Pasirinkite norimą dydį ir '82 ( Linux swap / Solaris )' tipą. Sukurtas skirstinys turėtų pasirodyti kaip *sda3*.

Paskutinis žingsnis - sukurti skirstinį, kuriame gulės /home direktorija. Pasirinkite *primary* ir '83 Linux' skirstinio tipą. Sukurtas skirstinys pasirodys kaip *sda4*.

Pavyzdžiui:

Name	Flags	Part Type	FS Type	[Label]	Size (MB)
sda1		Primary	Linux		15440 #root
sda2		Primary	Linux		10256 #/var
sda3		Primary	Linux swap / Solaris		1024 #swap
sda4		Primary	Linux		140480 #/home

Telieka tik išsaugoti skirstinių architektūra pasirinkus **Write** ir surinkti 'yes'. Po šitos operacijos *cfdisk* visiškai ištrins informaciją iš kietojo disko. Pasirinkite **Quit**, norint išeiti į programos. Pasirinkite *Done*, norint išeiti iš menu ir tęsti su *Direktorių pririšimas*

## 2.8.2 Direktorių pririšimas

Parinkite kiekvienam skirstiniui pririšama failų sistemos direktoriją ( Nepamirškite, jog skirstiniai baigiasi numeriais. Taip pat *sda* reiškia net ne skirstinį, o visą diską aplamai ).

**Failų sistemos tipai** Failų sistemos pasirinkimas yra labai asmeninis reikalas. Kiekvienas renkasi savo failų sistemą pagal jo poreikius. Kiekvienas failų sistemos tipas turi savo pliusų, minusų ir unikalumų. Trumpai galime aprašyti Arch Linux palaikomas failų sistemas:

1. **ext2** *Second Extended FileSystem* - sena, patikima GNU/Linux failų sistema. Labai stabili, bet be žurnalo palaikymo. Tikrai netinka / ( root ) ir /home direktorijoms, dėl labai ilgo fsck'o. Failų sistema *ext2* gali būti lengvai konvertuota į *ext3* failų sistemą. Dažniausiai *ext2* pasirenkama /boot direktorijai.
2. **ext3** *Third Extended FileSystem* - savo originale yra *ext2* failų sistema su žurnalo palaikymu. Failų sistema *ext3* turi atgalinį palaikymą *ext2*

failų sistemai. Ypač stabili, brandi ir kol kas plačiausiai naudojama failų sistema. Failų sistema buvo kuriama Gnu/Linux.

3. **ext4** *Fourth Extended FileSystem* - turi atgalinį *ext3* ir *ext2* palaikymą. Pristato diskų palaikymą, kurių talpa siekia virš 1 exabaito ir failo dydžių, kurių svoris siekia 16 terabaitų. Padidėjo ir subdirektorių palaikymas - nuo 32 000 ( *ext3* ) iki 64 000. Pat pat siūlo realaus laiko fragmentavimą.
4. **ReiserFS** (V3) - tai aukštos kokybės žurnalinė failų sistema, kuri naudoja labai neįprasta ir kūrybinga duomenų perdavimo algoritmą. ReiserFS dirba labai greitai, ypač jeigu kalba eina apie daug mažų bylų. ReiserFS yra greitai formatuojama, tačiau labai lėta pririšime. Pakankamai brandi ir stabili. ReiserFS šiuo metu nėra vystoma ( Reiser4 yra naujausia Reiser failų sistema ). Rekomenduojamą ReiserFS pasirinkti */var* direktorijai.
5. **JFS** - IBM sukurta **J**ournalled **F**ile**S**ystem yra pati pirma failų sistema, kuri turėjo žurnalo palaikymą. Iki prisijungimo prie Gnu/Linux IBM ilgus metus naudojo JFS savo AIX Operacinėje sistemoje. Šiuo metu JFS reikalauja mažiausiai sistemos resursų iš visų Gnu/Linux failų sistemų. Labai greita formatavime, pririšime ir gera visuose aspektuose, ypač jeigu ateina kirtis iš įvesties/išvesties įrenginio. Ne taip plačiai palaikoma kaip *ext* ar ReiserFS, tačiau labai brandi ir stabili sistema.
6. **XFS** - dar viena ankstyva žurnalo palaikymą turinti failų sistema. Ją sukūrė Silicon Graphics IRIX operacinei sistemai, o vėliau ji buvo prijungta prie Gnu/Linux. XFS siūlo labai greitą duomenų apskaitimą dirbant su dideliais skirstiniais arba didelėmis bylomis.

**Pastaba** JFS ir XFS sistemos negali būti sumažintos disko skaldymo programų ( kaip *gpart* ar *parted magic* ).

**Pastaba apie žurnalo palaikymą** Visos ( išskyrus *ext2* ) viršuje išvardintos failų sistemos turi žurnalo palaikymą. Tokios failų sistemos yra gan lanksčios klaidų atveju. Tai pasiekama labai paprastai - kai vartotojas padaro korekciją byloje - žurnalinė sistema pirmiausiai įrašo pakeitimo informaciją į savo žurnalą, o tik paskui pakeičia realią informaciją. Verta pastebėti, jog ne visos žurnalinės sistemos yra vienodos; pavyzdžiui tik *ext3* ir *ext4* sistemos siūlo *data-mode* žurnalo palaikymą. Tai reiškia, jog žurnalinė sistema savo žurnale saugo tiek duomenis, tiek meta duomenis ( tačiau tokiu atveju labai nukenčia sistemos greitis ). Kiti siūlo *ordered-mode*. Tokiu atveju žurnale saugomi tik meta duomenis. Tačiau bet kokių atveju, visos žurnalinės sistemos atstatys jūsų failų sistemą po lūžio. *data-mode* siūlo geriausią duomenų apsaugą, tačiau tai gali įtakoti sistemos greitį, kadangi duomenys yra rašomi du kartus - vieną kartą į žurnalą, o kitą kartą į diską. Remiantis kokio svarbumo duomenys bus saugomi diske, atitinkamai reikia pasirinkti ir disko failų sistemą.

**Tęsiame...** Pasirinkite ir sukurkite failų sistemą ( skirstinio formatavimas ) ir pririškite skirtinį prie / direktorijos, pasirinkdami **yes**. Toliau sistema paragins pridėti papildomus skirtinius. Mūsų pavyzdyje lieka *sda2* ir *sda4*. *Sda2* skirtiniui parenkame failų sistemą ir pririšame prie **/var**. Ir galiausiai parenkam *sda4* skirstiniui failų sistemą ir pririšame jį prie */home*.

**Pastaba** Jeigu nesukūrėte arba Jums tiesiog nereikia atskirto **/boot** skirstinio - ignoruokite pranešimą, kurį gausite apie **/boot** direktoriją. Grįžkite į pagrindinį menu.

## 2.9 Paketų pasirinkimas

- Core ISO: Savo paketų šaltiniu pasirinkite CD diską.
- Netinstall: Pasirinkite FTP/HTTP veidrodį. *Verta pažymėti, jog archlinux.org greitis yra sumažintas iki 50KB/s.*
- Visi įdiegiami paketai yra tik iš [core] saugyklos. Jie yra padalinti į dvi grupes:
  - **Base**: Paketai iš [core] saugyklos, kurie suteikia minimalią aplinką. Visuomet yra būtina pasirinkti šią kategoriją, tačiau galima pasirinkti, kokius paketus pašalinti.
  - **Base-devel**: Papildomi įrankiai iš [core] saugyklos: **make**, **auto-make** ir kt. Daugumai naujokų derėtų pasirinkti šią kategoriją, nes šitų įrankių anksčiau ar vėliau vistiek prireiks.

Po kategorijos pasirinkimo, jums bus pateiktas pilnas paketų sąrašas. Naudojamiesi tarpo klavišu galite pažymėti arba atžymėti paketą.

**Pastaba** Po kategorijos pasirinkimo, visi ten esantys paketai bus pažymėti įdiegimui.

**Pastaba** Jeigu yra reikalaujamas bevielis ryšis, prisiminkite pasirinkti įdiegimui **wireless\_tools** paketą. Kai kurie bevielio tinklo sąsajos taip pat reikalauja **ndiswrapper** ar/ir specifinės **firmware**. Jeigu bus naudojamas WPA kodavimas, taip pat reikalinga pasirinkti **wpa\_supplicant** paketą. Taip pat labai pravartu pasirinkti ir **netcfg**, kuris padės konfigūruojant tinklą ir profilius.

Po paketų pasirinkimo, palikite pasirinkimo langą ir tęskite toliau *Paketų įdiegimas*

## 2.10 Paketų įdiegimas

Toliau pasirinkite “Install Packages”.

- Netinstall: Dabar pacman paketų tvarkyklė atsiųs ir įdiegs paketus iš saugyklos. Eigą galima peržiūrėti Alt+F5, grįžti atgal į diegimo langą galima su Alt+F1.
- Core atvaizdas: pacman paketus ims iš CD/USB atvaizdo.

Kai kurios įdiegimo programos paklaus ar nenorite išsaugoti paketus į pacman programos podėlį. Jeigu pasirinksite ‘yes’, tuomet turite galimybę įdiegti programų senesnes versijas po jų atnaujinimų, todėl rekomenduojama yra pasirinkti ‘yes’. Ateityje bet kada bus galima ištrinti viską iš pacman programos

podėlio. Įdiegimo programa dabar turėtų įdiegti visus paketus, kaip ir Arch 2.6 branduolį į naują sistemą.

Po paketų parsisiuntimo, įdiegimo programa patikrins jų vientisumą. Toliau, jiniai sukurs branduolį iš parsisiųstų paketų.

## 2.11 Sistemos konfigūravimas

*Atidus sekimas ir supratimas yra kritiniai kriterijai, norint teisingai susikonfigūruoti sistemą.*

- Šiame žingsnyje yra galimas pagrindinės Arch Linux sistemos konfigūravimas.
- Ankstesnės įdiegimo versijos naudojo *hwdetect* surinkti informaciją apie esamą sistemą. Tokia technika jau pasenusi. Šiuo metu *udev* turėtų automatiškai atpažinti ir įkelti į branduolį reikalingus modulius sistemos krovimo metu.

Dabar bus suteikta galimybė pasirinkti redaktorių: *nano*, *joe* arba *vi* (*nano* yra pats lengviausias naudojime iš visų trijų). Po pasirinkimo, bus parodytas sąrašas bylų, kurias galima konfigūruoti.

**Pastaba** Šiame žingsnyje labai svarbu visus konfigūracinius failus pakoreguoti arba bent atidaryti ir įsitikinti, kad viskas gerai. Dažniausiai pasitaikanti klaida yra, kai vartotojai tiesiog praleidžia šitą žingsnį.

**Ar įdiegimo programa gali tai atlikti automatizuotai?** Slėpti sistemos konfigūravimą yra visiška “The Arch Way” priešingybė. Šiuo metu, naujausios Linux branduolio versijos ir aparatinės įrangos įrankiai siūlo tikrai gerą įrangos palaikymą ir automatinį konfigūravimą, tačiau Arch pateikia vartotojui visas galimas sistemos konfigūracijos galimybes, kadangi taip išlaikomas aiškumas ir sistemos resursų kontrolė. Kuomet baigsite redaguoti arba bent peržiūrėti konfigūracines bylas, jau turėsite bazinį supratimą apie Arch Linux sistemos rankinį konfigūravimą ir bazinę struktūrą. Tokia patirtis pravers naudojant ir sėkmingai palaikant sistemą po įdiegimo pabaigos.

### 2.11.1 /etc/rc.conf

Arch Linux naudoja */etc/rc.conf* kaip pagrindinę sisteminę konfigūracinę bylą. Byla sudaro didelę įvairovę konfigūraciniu nustatymu, kurie pagrinde reikalingi sistemos krovimosi metu. Kaip nusako ir pats bylos pavadinimas - konfigūracinė byla nusako visų */etc/rc\** bylų paleidimą, bei jų konfigūravimą.

#### LOCALIZATION sekcija

- **LOCALE=:** Šita eilutė nusako sistemos lokalę, kuri bus naudojama visų i18n programų ir įrankių. Galimų lokalių sąrašą galima peržiūrėti su *locale -a* komanda iš komandinės eilutės. Rekomenduojama palikti “en\_US”. Vėliau galima bus susigeneruoti dvi lokales - tiek lietuvišką, tiek angliską.

- **HARDWARECLOCK**=: Nusako koku formatu bus saugomas laikas įrenginyje, kuomet yra atliekama sinchronizacija pakrovimo ir stabdymo metu: *UTC* arba *localtime*. *UTC* yra labai geras, kadangi jis labai supaprastina visą laiko saugojimo procesą, tačiau, jeigu planuojate savo kompiuteryje turėti ir kitas operacinės sistemas, tokias kaip Windows, patartina yra pasirinkti *localtime*.
- **USEDIRECTISA**=: Nustatymu galima pasakyti, jog sistema naudos tiesioginį I/O kreipinį į įrangos laikrodį, o ne */dev/rtc*.
- **TIMEZONE**=: Nusako laiko juostą. Visas galimas laiko zonas galite peržiūrėti */usr/share/zoneinfo*.
- **KEYMAP**=: Nusako klaviatūros išdėstymą. Visus galimus klaviatūros išdėstymus galima rasti */usr/share/kdb/keymaps*. Šitas nustatymas veikia tik TTY aplinkoje, jis visiškai neįtakoja jokias langų tvarkyklės ar **X**.
- **CONSOLEFONT**=: Komandinės eilutės šriftas. Visus galimus komandinės eilutės šriftus galite peržiūrėti */usr/share/kdb/consolefonts*. Pagal numatytus nustatymus, tinka ir tuščias įrašas.
- **CONSOLEMAP**=: Nusako komandinės eilutės žemėlapi su *setfont* programa paleidimo metu. Galimus komandinės eilutės žemėlapius galima rasti */usr/share/kdb/consoletrans*, žinoma, jeigu tai yra būtina. Pagal numatytus nustatymus tuščias įrašas tinka.
- **USECOLOR**=: Pasirinkite “yes”, jeigu turite monitorių su spalvomis. :)

Byloje “LOCALIZATION” sekcija turi atrodyti panaši į:

```

LOCALE="en_US.utf8"
HARDWARECLOCK="localtime"
USEDIRECTISA="no"
TIMEZONE="Europe/Vilnius"
KEYMAP="us"
CONSOLEFONT=
CONSOLEMAP=
USECOLOR="yes"

```

#### **HARDWARE sekcija**

- **MOD\_AUTOLOAD**=: Jeigu nustatysite į “Yes”, tuomet naudosite **udev** automatiniam kompiuterio geležies aptikimui ir reikiamo modulio pakrovimui į sistemos branduolį. Jeigu nustatysite “No” viskas priklausys tik nuo Jūsų pačių savo kompiuterio žinių. Gali kilti būtinybė net kompiliuoti savo branduolį, modulius ir kt.
- **MOD\_BLACKLIST**=: Sekanti eilutė buvo pašalinta iš palaikymo sąrašo. Kaip alternatyva šiuo metu yra naudojama **MODULES**= eilutė, kuri nurodoma žemiau.
- **MODULES**=: Vartotojo nustatomi papildomi moduliai. Jeigu sistema turi floppy įrenginį, reikia pridėti “floppy”. Jeigu naudosite “loopback”

failų sistemas, tai galite pridėti "loop". Jeigu nenorite, jog tam tikras modulis pasikrautų automatiškai ( tai gali kilti, jeigu Jums reikės naudoti kitas tvarkykles, negu Jums užkrauna **udev** ), prie modulio pavadinimo pridėkite šauktuką (!). Tokiu atveju, **udev** bus priverstas nekrauti nurodyto modulio į branduolį.

Kaip pavyzdį galime pateikti užblokuotą *pcspeaker*:

```
MODULES=(!snd_pcsp !pcspkr loop)
```

### NETWORKING sekcija

- **HOSTNAME=**: Hostname galite pasirinkti tokį, koks Jums patinka. Tai yra Jūsų kompiuterio vardas tinkle. Kokį vardą bepasirinktumėt, svarbiausią, jį įtraukti ir į */etc/hosts*.
- **eth0=**: Sekantis punktas verčiasi kaip 'Ethernet, card 0'. Jeigu Jūs turite **statinį IP**, pridėkite prie sekančios eilutės savo IP adresą, netmask ir broadcast adresą. Jeigu Jūs naudojate **DHCP**, prie sekančios eilutės parašykite tik dhcp ( eth0="dhcp" ) ir Jums tinklas bus sukonfigūruotas automatiškai.
- **INTERFACES=**: Čia galite nurodyti savo sąsajas. Keletas sąsajų turi būti atskirtos tarpu, kaip pavyzdžiui:

```
(eth0 eth1 wlan0)
```

- **gateway=**: Jeigu naudojate **statinį IP**, nurodykite savo gateway adresą. Jeigu naudojate **DHCP** galite praleisti sekančią eilutę. Tačiau kai kurie vartotojai nurodė, jog ir naudojant DHCP, reikia nurodyti savo gateway. Pirma, pabandykite nieko nenurodyti, o jeigu prie tinklo kompiuteris vis tiek nesijungia - nurodykite savo gateway.
- **ROUTES=**: Jeigu naudojate **statinį IP**, pašalinkite ! prieš 'gateway'. Jeigu naudojate **DHCP**, galite palikti sekantį kintamąjį užkomentuotą. Tačiau, vėl gi, jeigu Jūsų kompiuteris neprisijungs prie interneto, naudokite statinio IP nustatymus.

Pavyzdys naudojant dinaminį IP ( **DHCP** ):

```
HOSTNAME="laptop"  
eth0="dhcp"  
INTERFACES=(eth0)  
gateway="default gw 192.168.0.1"  
ROUTES=(!gateway)
```

Pavyzdys naudojant **statinį IP**:

```
HOSTNAME="laptop"  
eth0="eth0 192.168.0.2 netmask 255.255.255.0 broadcast 192.168.0.255"  
INTERFACES=(eth0)  
gateway="default gw 192.168.0.1"  
ROUTES=(gateway)
```

Pakoreguokite `/etc/resolv.conf` nurodydami DNS serverius. Pavyzdžiui:

```
search my.isp.net
nameserver 192.168.1.1
nameserver 192.168.1.2
nameserver 192.168.1.3
```

Kai kurie procesai gali perrašyti `/etc/resolv.conf` turinį. Pavyzdžiui, pagal numatytus nustatymus Arch Linux naudoja **dhcpcd** DHCP klientą, kuris gali perrašyti bylos turinį, kai yra paleidžiamas. Kai kurie metodai gali išsaugoti nuo perrašymo sekančią bylą. Pavyzdžiui, galima nurodyti **dhcpcd** konfigūracinėje byloje, kad programa neperrašinėtų `/etc/resolv.conf` bylos turinį. Tam, tiesiog reikia pridėti `/etc/dhcpcd.conf` bylos turinio pabaigoje:

```
nohook resolv.conf
```

## DEAMONS sekcija

Sekantis masyvas yra paprastas skriptų sąrašas, kurie yra `/etc/rc.d` direktorijoje. Šioje eilutėje galima nurodyti kokie būtent skriptai bus paleisti sistemos krovimo metu ( taip pat ir jų eilė ). Asinchroninis paleidimas taip pat palaikomas ( tereikia prie skripto pridėti '@' ) ir jis labai paspartina sistemos krovimą.

```
DAEMONS=(network @syslog-ng netfs @crond)
```

- Jeigu prie skripto pavadinimo pridėti šauktuką (!), tai skriptas nebus paleistas.
- Jeigu prie skripto pavadinimo pridėti sraigę (@), tai skriptas bus paleistas asinchroniniu būdu. Paleidimo eilė nelauks, kol skriptas sėkmingai arba nesėkmingai bus užbaigtas, o tęs krovimą toliau.
- Jeigu į sistemą buvo įdiegtas naujas servisas ( tarkim 'httpd' ), galite pridėti jį prie DAEMONS sekcijos ir kiekvieną kartą sistema paleis 'httpd' ( apache2 ) jos krovimo metu.

## Apie DEAMONS

Deamons eilutė turi būti pakoreguota pagal kiekvieno vartotojo poreikius, tačiau, dabar būtina truputi pakalbėti apie daemons'us.

Daemon yra programa, kuri yra vykdoma šešėlyje ir kuri laukia pertraukties. Kai įvyksta pertrauktis - daemon siūlo savo sprendimus pertraukčiai apdoroti. Geras pavyzdys yra web serveris, kuris laukia, kol į jį bus padaryta kreiptis ( 'httpd' ) arba ssh serveris, kuris laukia, kol į jį bus padaryta kreiptis ( 'sshd' ). Tokios programos yra pilnos, tačiau yra programų, kurių darbą ne taip lengva pastebėti. Kaip pavyzdį galima nurodyti programą, kuri rašo visus pranešimus į sisteminių žurnalą ( 'syslog', 'metalog' ) ir programos, kurios suteikia grafinį priėjimą prie sistemos ( 'gdm', 'kdm' ). Visos sekančios programos gali būti įtrauktos į daemon sąrašą ir paleistos sistemos krovimo metu. Vėliau pateiksime tam tikras, naudingas daemon programas.

Istoriškai, terminas *daemon* prigijo dėka MIT projekto MAC programuotojų. Jie pasiskolino pavadinimą iš *Maxwell demon*, įsivaizduojamą sutvėrimą iš

ižymaus ir sudėtingo eksperimento, kuris visuomet dirbą šešėlyje, rikiuodamas molekules. \*nix sistemos pasisavino šitą terminą ir pakeitė jo kilmę kaip į žodžių junginį “**disk and execution monitor**”.

## 2.11.2 /etc/fstab

**Fstab** ( trumpinimas iš žodžių **file systems table** ) yra dalis sistemos konfigūracijos, kuriame nurodomi visi galimi diskų ir disko skirstinių elgsena sistemoje, t.y. prie kurios direktorijos ir kokiomis teisėmis turi būti pririšti diskai, bei jų skirstiniai. Byla */etc/fstab* dažniausiai naudojama **mount** programos. Mount komanda paima disko skirstinio failo sistemą ir prijungia ją prie sisteminės direktorių hierarchijos. Paleidimo metu, kuomet praeina 3/4 viso paleidimo, **mount -a** komanda perskaito */etc/fstab* bylą ir pagal perskaitytus nustatymus prijungia nurodytus skirstinius prie direktorių hierarchijos. Jeigu prie skirstinio yra parašyta *noauto*, sistemos krovimo metu **mount -a** skirstinio niekur neįjungs.

### /etc/fstab pavyzdys

# <file system>	<dir>	<type>	<options>	<dump>	<pass>
none	/dev/pts	devpts	defaults	0	0
none	/dev/shm	tmpfs	defaults	0	0
/dev/sda1	/	jfs	defaults,noatime	0	1
/dev/sda2	/var	reiserfs	defaults,noatime,notail	0	2
/dev/sda3	swap	swap	defaults	0	0
/dev/sda4	/home	jfs	defaults,noatime	0	2

- **<file system>**: nusako disko skirstinį arba nutolusi failų sistemą su kuria bus dirbama.
- **<dir>**: nusako direktoriją, prie kurios bus pririštas anksčiau nurodytas disko skirstinys arba nutolusi failų sistemą. Jeigu skirstinys yra *swap* tipo, čia turi būti nurodytas *'swap'*. Verta pastebėti, jog *swap* skirstiniai realiai nėra prijungiamos prie direktorių hierarchijos.
- **<type>**: nusako failų sistemos tipą. Linux branduolys palaiko labai daug failų sistemų. Galite patikrinti kokias failų sistemas palaiko sisteminis branduolys, peržiūrėję */proc/filesystems* direktoriją. Įrašas *'swap'*, nurodo, jog bus dirbama su *swap* tipo skirstiniu ar byla. Įrašas *'ignore'* nurodo, jog sekanti eilutė turi būti ignoruojama. Toks įrašas naudingas tuomet, kai tenka laikinai atjungti kokį nors išorinį ar vidinį diską.
- **<options>**: nusako parinktis, kurios bus įvykdytos prijungimo metu. Eilutė formatuojama kaip parinkčių sąrašas, kur kiekviena parinkti atskiriama kablelių, be jokių tarpų. Pagrinde, eilutė turi susidėti iš prijungimo tipo ir dar papildomos parinktys, kurios priklauso nuo failų sistemos tipo. Detalesnei dokumentacijai peržiūrėkite *'man mount'*.
- **<dump>**: nusako kokia failų sistema turi būti perkelta į *'savartyną'*. Dump yra savotiška atsarginių kopijų darymo tarnyba. Jeigu penktas laukas nėra nurodomas, skirstinys nurodomas kaip nereikalaujantis atsarginių kopijų darymo.



- **<pass>**: nusako failų sistemos tikrinimo prioritetą sistemos krovimosi metu. Pagrindinė failų sistema turi būti nurodoma 1, o visos kitos turi turėti nurodymą 0 arba 2. Failų sistemos, kurios yra tam pačiam diske, bus tikrinamos sinchroniškai. Failų sistemos, kurios yra skirtinguose diskuose, bus tikrinamos tuo pačiu metu naudojant paralelizmą, esantį techninėje įrangoje. Jeigu šeštas parametras nėra nurodomas arba yra 0, fsck netikrins failų sistemos.

### 2.11.3 /etc/mkinitcpio.conf

*Daugumai vartotojų nereikia redaguoti sekančios konfigūracinės bylos, tačiau galite perskaityti nurodytą informaciją savo žinių gilinimui.*

Konfigūracinė byla leidžia toliau testuoti kruopštų sistemos nustatymą ties ram failų sistema arba **initramfs** arba istoriškai vadinamą **initrd**. **Initramfs** yra gzip suarchyvuotas sistemos atvaizdas, kuris gali būti tik skaitomas sistemos krovimo metu. **Initramfs** tikslas yra suteikti tiltą sistemai iki pagrindinės failų sistemos. Tai reiškia, kad jis turi užkrauti visus reikiamus modulius į sisteminių branduolį. Tuo metu, kai **initramfs** užkrauna reikiamus modulius automatiškai, arba rankiniu būdu, viskas toliau perduodama sisteminiam branduoliui. Sekant iš **initramfs** paskirties, byloje turi būti nurodomi tik moduliai, kurie yra reikalingi norint sėkmingai pasiekti sistemos branduolį ir pagrindinę failų sistemą. Jam nereikia nurodyti kiekvieno modulio, kuris bus naudojamas sistemos. Likusi dalis modulių bus užkrauta vėliau, kartu su branduoliu, **udev** pagalba, **init** metu.

**mkinitcpio** yra sekanti karta **initramfs** kūrime. Jis turi labai daug privatumu, lyginant su senais **mkinitrd** ir **mkinitramfs** skriptais.

- Jis naudoja **glibc** ir **busybox**. Taip suteikiama maža ir lengva bazė.
- Jis naudoja **udev** krovimo metu taip pašalindamas modulius, kurie nėra reikalingi norint pasiekti sisteminių branduolį, bei pagrindinę failų sistemą.
- Jo kablo tipo paleidimo skriptai ( **hooks** ) yra lengvai praturtinami su papildomais kabliais, kurie įtraukiami per **pacman** programą. Taip visiškai nereikia redaguoti **mkinitcpio** bylos.
- Jis palaiko **lvm2**, **dm-crypt**, **raid**, **swsusp** bei **suspend2** pažadinant ar užmigdant ir krovimą iš usb įrangos.
- Daugelis ypatybių gali būti pakeisti per sistemos branduolio komandinę eilutę, nereikalaudant papildomai kompiliuoti atvaizdo.
- **Mkinitcpio** skriptas laidžia įtraukti sisteminių branduolį į atvaizdą bei tiesiog būti sisteminiam branduolyje.
- Jo lankstumas daugeliu atveju nereikalauja perkompiliuoti sisteminio branduolio.

Naudojant **RAID** arba **LVM** pagrindinėje failų sistemoje, turi būti sukonfigūruoti tam tikri kabliai ( **HOOKS** ). Pridėkite “**usbinput**” jeigu naudojate usb klaviatūrą. Nepamirškite pridėti “**usb**”, jeigu diegiate Arch Linux į usb laikmeną, kaip pavyzdžiui:

```
HOOKS="base udev autodetect pata scsi sata usb filesystems keymap usbinput"
```

Jeigu yra būtinas USB įrenginių, FireWire, PCMCIA, NFS, RAID, LVM2, užkoduotų laikmenų ar DSDT palaikymas - atitinkamai sukonfigūruokite kablius ( HOOKS ).

**Mkinitcpio** yra Arch inovacija, kuri buvo kuriama Aaron Griffin ir Tobias Powalowski su tam tikra pagalba iš bendruomenės.

#### 2.11.4 /etc/modprobe.d/modprobe.conf

Sekanti konfigūracinė byla gali būti naudojama papildomiems nustatymams branduolio moduliams. Nėra reikalavimo redaguoti sekančią bylą.

#### 2.11.5 /etc/resolv.conf ( statiniam IP )

Resolver'is yra sąrašas kelių, kurie yra aprašyti C biblioteka. Jie suteikia priėjimą prie interneto domain name systems ( DNS ). Viena iš pagrindinių DNS užduočių yra konvertuoti domeno vardą į ip adresą. Taip internetas padaromas žymiai draugiškesnis vartotojui. Resolverio konfigūracinėje byloje, */etc/resolv.conf* pateikiama informacija apie adresus, kuriais resolveris vadovaujasi pirmiausiai.

Jeigu naudojate statinį IP, turite nurodyti savo DNS serverius. Tarkim, jeigu naudojate OpenDNS:

```
nameserver 208.67.222.222
nameserver 208.67.220.220
```

Jeigu naudojate maršrutizatorių, Jums reikia nurodyti DNS'us jame, o resolverio konfigūracinėje byloje tik padaryti nukreipimą iki maršrutizatoriaus.

```
nameserver 192.168.1.1
```

Jeigu naudojate DHCP, taip pat galite nurodyti DNS serverius, tačiau tai nėra būtina, kadangi visos operacijos atliekamos automatiškai.

#### 2.11.6 /etc/hosts

Sekanti konfigūracinė byla susieja IP adresus su tinklo vardais, bei trumpiniais. Kiekviena konfigūracinė eilutė turi savyje turėti tokią informaciją:

```
<IP-adresas> <tinklo-vardas> [trumpiniai..]
```

Pridėkite savo tinklo vardą, priklausomai nuo */etc/resolv.conf* nurodytam vardui:

```
127.0.0.1 localhost.localdomain localhost tinklo-vardas
```

Jeigu naudojate statinį IP, pridėkite dar vieną eilutę, kuri susidėtų iš:

```
192.168.1.100 tinklo-vardas.domain.org tinklo-vardas
```

### 2.11.7 /etc/hosts.deny ir /etc/hosts.allow

Pakoreguokite sekančias konfigūracines bylas, jeigu tai yra būtina. Toks poreikis gali iškilti tuomet, jeigu naudosite ssh. Pagal numatytą konfigūraciją, kiekvienas į sistemą ateinantis prašymas prisijungti prie jūsų kompiuterio bus atmestas. Pakoreguokite */etc/hosts.allow* bylą su tam tikrais parametrais:

- kiekvienas gali prisijungti

```
sshd: ALL
```

- apriboti prisijungimą tik iš vieno IP

```
sshd: 192.168.0.1
```

- apriboti prisijungimą tik iš vietinio LAN tinklo

```
sshd: 192.168.0.
```

- apriboti prisijungimą tik iš IP sąrašo

```
sshd: 10.0.0.0/255.255.255.0
```

Jeigu neplanuojate naudoti ssh, palikite sekančią konfigūracinę bylą nekeistas. Tai tik pridės sistemai saugumo.

### 2.11.8 /etc/locale.gen

Komanda **/usr/sbin/locale-gen** perskaito duomenis iš */etc/locale.gen* ir sugeneruoja specifines lokales. Lokalės naudojamos **glibc** ir bet kokia kita lokale programa ar biblioteka teksto atvaizdavimui.

Pagal numatytus nustatymus, byla lieka tuščia, su užkomentuota dokumentacija. Konfigūracija atliekama tik vieną kartą. **locale-gen** yra paleidžiamas kiekvieną kart, kai tik yra atnaujinamas **glibc**, generuojamas visas reikiamas lokales, kurios yra aprašytos */etc/locale.gen* byloje.

Pasirinkite reikiamas lokales pašalindami komentaro ženklą (#) eilutės pradžioje, pavyzdžiui:

```
en_US ISO-8859-1
en_US.UTF-8
```

Įdiegimo programa paleis locale-gen skriptą ir sugeneruos reikiamas lokales. Tolimesnėje sistemos naudojime, galite keisti lokales konfigūracinę bylą ir sugeneruoti lokales su **locale-gen** komanda. Būtina sąlyga, jog skriptas turi būti paleistas su root vartotojo teisėmis.

### 2.11.9 Pacman veidrodis

Pasirinkti reikiamą **pacman** veidrodį.

Šiuo metu lietuviškų pacman veidrodžių nėra.

- Oficialus *archlinux.org* veidrodis yra limituotas iki 50 KBps greičio, tad, jeigu turite greitesnį internetą ir nemėgstate laukti - rekomenduojama pasirinkti kitą veidrodžio šaltinį.

Knygos rašymo metu, Arch Linux Lietuva derina veidrodžio paleidimo galimybes VGTU Elektronikos fakultete.

### 2.11.10 Root slaptažodis

Slaptažodžio pasirinkimas root vartotojui turi būti labai atsakingas darbas. Reikia jį pasirinkti kiek įmanoma kompleksinį, tačiau kombinacija turi likti įmanoma atsiminti.

Grįžkite į pagrindinį menu ir tęskite su bootloader įdiegimu.

### 2.11.11 Pabaiga

Kuomet pasirinksite “Done”, sistema perkompiliuos visus reikiamus atvaizdus, remiantis konfigūracine informacija. Šitas procesas gali šiek tiek užtrukti. Kuomet viskas bus užbaigta, įdiegimo programa perkels į pagrindinį menu.

## 2.12 Bootloader įdiegimas

Kadangi įdiegimas vyksta tuščioje sistemoje, mums prireiks sistemos bootloader (lietuviškai tai verčiama kaip ‘paleidyklė’). GNU GRUB yra rekomenduojamas pasirinkimas. Kaip alternatyva, galima pasirinkti Lilo arba Syslinux.

### 2.12.1 GRUB

Numatyti GRUB nustatymai, kurie yra */boot/grub/menu.lst*, turėtų tenkinti kiekvieno pradedančiojo vartotojo poreikius ir sistema tikrai pasikraus, jeigu paliksite konfigūracinę bylą nepaliesta. Tačiau būtina patikrinti ar visi nustatymai yra teisingi. Konfigūravimo darbus ir eksperimentus su sekančią bylą galima nukelti į po-sistemos įdiegimo darbus.

Įvykiams užbėgant už akių, prie *kernel* eilutės galima pridėti *vga=<numeris>* eilutę. Sekanti opcija, leis terminalui dirbti didesnėje raiškoje ir tai palengvina vartotojo darbą su terminalu. Lentelė su numeriais ir ekrano raiška yra pateikiamos */boot/grub/menu.lst* byloje.

Pavyzdys:

```
title Arch Linux (Main)
root (hd0,0)
kernel /boot/vmlinuz26 root=/dev/sda1 ro
initrd /boot/kernel26.img
```

Pavyzdys, kuomet */boot* direktorija yra atskirame skirstinyje:

```

title Arch Linux (Main)
root (hd0,0)
kernel /vmlinuz26 root=/dev/sda3 ro
initrd /kernel26.img

```

Paiškinimai:

- Eilutė 1: *title* yra tekstas, kuris bus rodomas kaip menu pasirinkimas. “Arch Linux (Main)” bus pavaizduotas menu kaip galimas pasirinkimas.
- Eilutė 2: *root* GRUB root skirstinys. Tai yra diskas arba disko skirstinys, kur galima rasti */boot* direktoriją. Arba tiksliau, kur galima rasti GRUB’o stage2 bylą. GRUB numeracija prasideda nuo 0, ir naudoja hdx,y formatą. Pavyzdyje, */boot* guli pirmame diske, pirmame skirstinyje, arba (hd0,0)

- Eilutė 3: *kernel*. Sekanti eilutė nurodo:

- *kernel* ( sistemos branduolio ) kelią ir pavadinimą, priklausomai nuo GRUB’o root.

Pavyzdyje, */boot* ir **vmlinuz26** yra tame pačiame skirstinyje, kaip ir pagrindinė failų direktorija, tai absoliutus kelias iki sisteminio branduolio ( *kernel* ) yra **/boot/vmlinuz26**.

Kuomet */boot* yra atskirame skirstinyje, absoliutus kelias iki sisteminio branduolio būtų tiesiog **/vmlinuz26**.

- *root=* argumentas nusako disko skirstinį, kuriame yra pagrindinė bylų direktorija (/). Arba tiksliau, tai disko skirstinys, kuriame galime rasti */sbin/init*.
- Įsiminti sekančių argumentų tvarka galima taip: pirmas argumentas nusako GRUB’ui kur galima rasti sisteminį branduolį, o antras argumentas nusako sisteminiam branduoliui kur galima rasti pagrindinę bylų direktoriją.
- Paskutinė argumentas nusako sisteminio branduolio opcijas.

Mūsų pavyzdys, **ro** reiškia, kad sistemos krovimo metu, bylos sistema yra prijungiama prie sistemos tik su skaitymo teisėmis.

- Eilutė 4: **initrd** Kelias ir bylos pavadinimas vidinei RAM bylų sistemai, priklausomai nuo GRUB’o pagrindinės direktorijos. Pavyzdyje, viskas vyksta tam pačiam diske ir tam pačiam skirstinyje, taigi pilnas kelias yra */boot/kernel26.img*.

Jeigu viskas vykėtų atskiruose diskuose, ar skirstiniuose, absoliutus kelias būtų tiesiog */kernel26.img*.

Įdėkite GRUB sistemos bootloaderį į pagrindinį disko krovimo parametrą ( pavyzdyje tai yra sda ).

## 2.13 Perkrovimas

Štai ir viskas. Bazinės sistemos konfigūravimas ir įdiegimas užbaigtas. Išeikite iš įdiegimo programos ir perkraukite savo kompiuterį:

**# reboot**

Kuomet sistema baigs išjungimo operacijas - išimkite įdiegimo diską arba įdiegimo laikmeną.

## skyrius 3

# Pagrindinės sistemos atnaujinimas ir konfigūravimas

Šviežia sistema pradės krautis ir krovimas bus užbaigtas tuomet, kai pasirodys prisijungimo sąsaja ( prompt ).

**Sveikiname, ir sveiki atvykę į naują Arch Linux pagrindinę sistemą!**

Nauja Arch Linux sistema ir pilnavertė GNU/Linux aplinka, kuri yra paruošta konfigūravimui. Nuo šio žingsnio, Jūs esate pats savo sistemos kalvis. Jūsų sistema bus tokia, kokia Jums yra reikalinga.

Prisijunkite kaip root vartotojas. Toliau tęsime su pacman ir sistemos atnaujinimu.

**Pastaba** Yra galimos virtualios konsolės. Jas galite pasiekti per Alt+F1 ... Alt+F6.

### 3.1 Žingsnis 1: Tinklo konfigūravimas

*Sekantis skyrius bus jums vadovo pozicijoje konfigūruojant tinklą, jeigu jis jums neveikia.*

Jeigu įdiegimo metu teisingai buvo atliktas tinklo konfigūravimas, tinklas jums turėtų veikti. Tai galima patikrinti su ping komanda:

```
ping -c www.google.lt
```

*Jeigu tinklas yra pajungtas, tęskite su Žingnis 2.*

Jeigu po ping bandymo ateina pranešimas apie “unknown host”, tinklas nėra teisingai sukonfigūruotas. Pirmą, ką reiktų padaryti, tai dar kartą patikrinti konfigūracines bylas:

- **/etc/rc.conf** Ypač HOSTNAME= bei NETWORKING sekcijas.
- **/etc/hosts** Patikrinkite nustatymų formatą.
- **/etc/resolv.conf**, jeigu naudojate statinį IP adresą. Jeigu naudojate DHCP, byla yra sukuriamą ir ištrinama dinamiškai.

### 3.1.1 LAN tinklas

Patikrinkite savo tinklo sąsajas su

```
# ifconfig -a
```

Išvykdžius šią komandą, bus pateiktos visos tinklo sąsajos, kurias sistema atpažino. Turėtumėt pamatyti kažką panašaus į *eth0* arba *eth1*.

- Statinis IP

Jeigu yra būtina, galima nustatyti savo statinį IP:

```
# ifconfig eth0 <ip adresas> netmask <netmask> up
```

Ir numatytą gateway su:

```
# route add default gw <gateway-ip-adresas>
```

Patikrinkite ar */etc/resolv.conf* byloje yra įrašyti DNS serveriai ir pridėkite juos, jeigu jų nėra. Patikrinkite savo tinklą su ping komanda ir jeigu tinklas atsirado - įveskite viską į savo *rc.conf* bylą, kaip aprašyta aukščiau, pirmame skyriuje.

- DHCP

Jeigu naudojate DHCP, pabandykite tiesiog:

```
# dhcpcd eth0
```

Jeigu tai suveiks - įveskite reikiamus konfigūracijos parametrus į */etc/rc.conf*, kaip nurodyta pirmoje dalyje.

### 3.1.2 Belaidis tinklas

- Patikrinkite ar reikiama tvarkyklė sukūrė naudojamą bevielio tinklo sąsają:

```
# iwconfig
```

Komanda turėtų parodyti esamas bevielio tinklo sąsajas. Jo pavadinimai gali būti *wlan0*, *wlan1* arba *eth1*.

- Įjunkite norimą sąsają su *ifconfig <sąsaja>*, pavyzdžiui:

```
# ifconfig wlan0 up
```



**Pastaba** Jeigu išmes klaidos pranešimą 'SIOCSIFFLAGS: No such file or directory', tai reiškia, jog tinklo plokštei trūksta firmware'o.

- Susiekite norimą tinklą su sąsaja. Sekantis procesas priklauso nuo tinklo tipo, prie kurio norima prisijungti, tačiau pagrindinis reikalavimas yra žinoti bevielio tinklo ESSID, t.y. tinklo pavadinimą ( pavyzdžiui 'eduroam' ):
  - Pavyzdys, naudojant neužkoduotą tinklą:  

```
# iwconfig wlan0 essid "eduroam"
```
  - Pavyzdys, naudojant WEP kodavimą ir šešioliktinį rakta:  

```
# iwconfig wlan0 essid "eduroam" key 0241baf34c
```
  - Pavyzdys, naudojant WEP ir ASCII rakta:  

```
# iwconfig wlan0 essid "eduroam" key s:secret-pass
```
  - Naudojant WPA, prisijungimas prie tinklo reikalauja truputi daugiau pastangų:  

```
# wpa_passphrase eduroam "slaptazodis" > /etc/wpa_supplicant.conf  
# wpa_supplicant -B -Dwext -i wlan0 -c /etc/wpa_supplicant.conf
```

**Pastaba** Po -D esantis argumentas 'wext' yra tvarkyklės pavadinimas, kuris yra naudojamas sąsajai su bevieliu tinklu.

Patikrinkite ar tinklas teisingai sukonfigūruotas:

```
# iwconfig wlan0
```

- Kreiptis į bevieli tinklą atliekama paprasčiausiai su *dhcpcd* <sąsaja>:

```
# dhcpcd wlan0
```

- Patikrinkite ar tinklas sukonfigūruotas teisingai:

```
# ping -c www.google.lt
```

Jeigu viskas buvo atlikta teisingai - tinklas turėtų veikti.

### 3.1.3 Proxy serveris

Jeigu esate už proxy serverio, pakoreguokite */etc/wgetrc* įgalindami *http\_proxy*, bei *ftp\_proxy*.

### 3.1.4 Analoginis modemas, ISDN ir DSL (PPPoE)

Analoginis modemas

ISDN

DSL

## 3.2 Žingsnis 2: Atnaujinimas su pacman

Dabar, kai turime veikiantį priėjimą prie tinklo, galime atnaujinti savo sistemą, pasitelkdami pacman.

### 3.2.1 Kas yra pacman ?

Pacman yra Arch Linux paketų tvarkyklė ( **package manager** ). Pacman yra parašytas su C kalba ir nuo pagrindų yra projektuotas kaip lengva, greita ir mažai darbinės atminties reikalaujanti programa. Pacman prižiūri kiekvieną paketą, kuris yra įdiegtas į sistemą: atlieka įdiegimą, pašalinimą, atnaujinimą, rankinių būdų sukompiliuotų paketų įdiegimą, priklausomybių tikrinimą, nuotolinę, bei vidinę paketų paiešką ir kita. Pacman programos pateikiama informacija yra suprantama ir suteikia ETA kiekvieno paketo atsisiuntimo metu. Arch naudoja pkg.tar.gz archyvus ir šiuo metu yra vykdomi darbai pereinant prie pkg.tar.xz formato.

Dabar pacman bus naudojamas atsisiunčiant paketus iš paketų saugyklos ir diegimui į sistemą.

### 3.2.2 Paketų saugyklos

Dabartiniu metu pacman siūlo tokias paketų saugyklos galimybes:

**[core]**

Paprastu principu remiantis, [core] suteikia prieigą tik po vieną įrankį, skirtą atlikti tam tikrą pagrindinės Arch Linux sistemos darbą. GNU toolchain, sisteminis branduolys ( Linux Kernel ), vienas redaktorius, vienas komandinės eilutės klientas ir kt. ( tačiau yra ir keletas išimčių, kaip pavyzdžiui vi ir nano yra pasiekiami [core] saugykloje ). Saugykloje yra tie pagrindiniai paketai, kurie turi būti įdiegti į bet kokią veikiančią sistemą, norint užtikrinti, kad sistema veiktų stabiliai ir patikimai. Tai yra visiškai sistemiškai kritiniai paketai.

- Pagrindinės sistemos programuotojų palaikomi
- Visi paketai yra binariniai
- Prieinami per pacman
- Core sistemos diegimo atvaizdas susideda vien iš [core] saugykloje saugomų paketų.

### [extra]

Saugykloje [extra] saugomi visi paketai, kurie nėra būtini bazinėje Arch Linux sistemoje, tačiau jie suteikia pilną Arch Linux darbo aplinką, tad be jų neišsiversti. X, KDE, bei Apache yra saugomas sekančioje saugykloje.

- Pagrindinės sistemos programuotojų palaikomi
- Visi paketai yra binariniai
- Prieinami per pacman

### [testing]

Sekančia saugykla patartina naudotis tik labai patyrusiems vartotojams, todėl šiame vadove jos neapžvelgsime.

### [community]

Saugykla yra prižiūrima Arch Linux patikimų vartotojų ( *TU* ) ir paprasčiausiai yra binarinis AUR variantas. Jame saugomi binariniai paketai, kurie yra suorganizuoti PKGBUILD pagalba iš AUR, kurie sulaukė daugiausiai balsų ir buvo priimti į saugyklą. Kaip ir kiekviena saugykla, išvardinta aukščiau, [community] galima pasiekti pacman pagalba.

- TU palaikomi
- Visi paketai yra binariniai
- Prieinami per pacman

### [multilib]

Vartotojai, kurie naudoja 64 bitų architektūrą turi poreikį diegti paketus, kurie nėra pritaikyti tokiai architektūrai. Paprasti 32 bitų paketai gali būti naudojami 64 bitų versijoje, tačiau sistemoje turi egzistuoti specifinės bibliotekos, kurios ir yra saugomos [multilib] saugykloje.

- Pagrindinės sistemos programuotojų palaikomi
- Paketai yra binariniai
- Prieinami per pacman

**Pastaba** Norint naudotis saugykla, reikia pridėti kelias eilutes į */etc/pacman.conf*

```
[multilib]
```

```
Include = /etc/pacman.d/mirrorlist
```

### AUR (nepalaikomas)

AUR saugomi visi oficialiai nepalaikomi paketai, kurie negali būti pasiekiami per pacman. AUR'e nėra jokių binarinių paketų. Tuo tarpu, AUR suteikia daugiau nei šešiolika tūkstančių PKGBUILD skriptų, kurių pagalba galima labai lengvai sukompiliuoti bet kokią norimą paketą iš pradinio kodo. Kuomet AUR PKGBUILD skriptas surenka pakankamai balsų, jo binarinis variantas yra perkeliamas į [community] saugyklą. Tam taip pat turi pritarti ir TU.

- TU palaikomi
- Visi PKGBUILD yra skriptai
- Pagal numatytus nustatymus, neprieinamas per pacman

Sekančią saugyklą galima pasiekti kitais būdais, t.y. pasitelkiant pacman apgaubtu ( wrapper ).

### 3.2.3 /etc/pacman.conf

Kiekvieno pacman paleidimo metu, jis skaitys */etc/pacman.conf*. Konfigūracinė byla yra suskaldyta į du blokus arba saugyklas. Kiekvienas blokas nusako paketų saugyklą, kurią pacman gali naudoti paketų paieškai. Išimtis yra opcijų blokas, kuris nustato globalius kintamuosius.

Verta pastebėti, jog pagal numatytus nustatymus viskas turi veikti, taigi bylos redagavimas šiame žingsnyje nėra būtinas, tačiau patikrinimas yra rekomenduojamas.

```
# nano /etc/pacman.conf
```

Ijunkite visas norimas saugyklas, pašalindami komentaro (#) ženklą eilutės pradžioje.

**Pastaba** Įsitikinkite, jog atkomentavote tiek saugyklą, kurios pavadinimas apskliaustas laužtiniais skliaustais ([saugyklos-pavadinimas]), tiek ir *Include* eilutę, kuri seka iškart po saugyklos pavadinimo. Tai dažna klaida.

### 3.2.4 /etc/pacman.d/mirrorlist

Konfigūracinė byla nusako saugyklų veidrodžius bei prioritetus.

#### **Veidrodžių sąrašo generavimas pasitelkiant rankmirrors skriptą ( nebūtina )**

Arch Linux projekte yra */usr/bin/rankmirrors* skriptas, kuris bando išrinkti archiausiai esančius veidrodžius, kurie yra atkomentuoti */etc/pacman.d/mirrorlist* byloje. Greitesni veidrodžiai dramatiškai pakeis pacman darbą, bei absoliutų Arch Linux naudojimą. Skriptas gali būti paleidžiamas periodiškai, turint galvoje, kad veidrodžių apkrova laikui bėgant kinta. Reikia pridurti, kad *rankmirrors* skriptas netestuoja veidrodžių greitį. Tačiau tai galima patikrinti su kito tipo įrankiais - *wget* arba *rsync*. Galiausiai, po visų testų, galima surinkti geriausių veidrodžių rinkinį į */etc/pacman.d/mirrorlist*.

### Priverstinis pacman paketų atnaujinimas

Ivykdykite sekančią komandą:

```
# pacman -Syy
```

Siunčiant du *-refresh* arba *-y* argumentus verčia pacman atnaujinti visus paketus, nors sistemos duomenimis, jie yra naujausios versijos. Vykdyti *pacman -Syy* kiekvieną kartą, kuomet yra keičiamas veidrodis, yra gera praktika. Taip galima išvengti labai daug galvos skausmų.

Įdiegite curl, pasitelkiant pacman:

```
# pacman -S curl
```

Nueikite iki */etc/pacman.d* direktorijos:

```
# cd /etc/pacman.d/
```

Padarykite atsarginę *mirrorlist* kopiją:

```
# cp mirrorlist mirrorlist.backup
```

Pakoreguokite *mirrorlist.backup* ir atkomentuokite visus veidrodžius, kurie geografiškai yra arčiausiai jūsų kompiuterio

```
# nano mirrorlist.backup
```

Dabar teliko tik paleisti *rankmirrors* skriptą, kuris skaitytų *mirrorlist.backup* bylą su *-n* argumentu ir perduoti visą srautą į naują *mirrorlist* bylą:

```
# rankmirrors -n 6 mirrorlist.backup > mirrorlist
```

**Pastaba** *-n 6* nusako parinkti šešis arčiausius veidrodžius.

Priverstinai paleiskite pacman atnaujinti visus paketus:

```
# pacman -Syy
```

### 3.2.5 Mirrorcheck sistemos naujumui tikrinimui

Kai kurie oficialūs veidrodžiai gali turėti ir pasenusių paketų. Arch Linux mirrorcheck praneša apie įvairius aspektus, kurie gali kilti veidrodžiams - tinklo problemos, duomenų surinkimo problemos, paskutinis paketų atnaujinimo laikas ir kt.

Gera yra tikrinti savo */etc/pacman.d/mirrorlist* ir stebėti, ar nurodyti veidrodžiai yra naujausi.

### 3.2.6 Paketų atnaujinimo ignoravimas

Po *pacman -Syu* kiekvienas sistemos paketas bus atnaujintas. Tačiau galima ir priversti *pacman* neatnaujinti tam tikrų paketų. Tipinė situacija yra ta, jog paketo atnaujinimas tiesiog sustabdys stabilų sistemos veikimą. Tokiu atveju yra du pasirinkimai: nurodyti *pacman* kokius paketus praleisti su *-ignore* argumentu arba laikinai nurodyti praleidžiamus paketus */etc/pacman.conf* byloje, *IgnorePkg* masyve. Išvardinkite kiekvieną paketą eilute, palikdami tarpą tarp kiekvieno paketo:

```
IgnorePkg = wine vim nano
```

Tipinis būdas naudoti Arch yra atnaujinti visus paketus, nebent kažkokio paketo atnaujinimas dar nėra galimas. Tokiu atveju galima pasinaudoti ABS. Daugelis vartotojų sukurti kompiliavimo skriptai yra pasiekiami per AUR ir gali būti įdiegti rankiniu būdu, pasitelkiant *makepkg*.

Pagrindiniai vartotojai viliasi, jog visa sistema atsinaujins paprasčiausiai paleidžiant *pacman -Syu*, nenurodant kokį būtent paketą atnaujinti nereikia. Tačiau kiekvienas vartotojas gali būti skirtingas ir tuo labiau - kiekvieno vartotojas turi savo pasirinkimo laisvę. Yra didelė tikimybė, kad įtraukus kritinio paketo atnaujinimo ignoravimą, sistema gali veikti nestabiliai arba visai nutraukti savo darbą. Dažniausiais, blogas sistemos veikimas ir nutinka, kai vartotojai įtraukia reikiamus paketus į atnaujinimo ignoravimo sąrašą.

Paketų atnaujinimo ignoravimas yra labai suktas dalykas ir pradedančiajam jo naudoti nėra patartina.

### 3.2.7 Konfigūracinių bylų ignoravimas

Tokiu pačiu principu galite apsaugoti ir savo konfigūracines bylas nuo *pacman -Syu* atnaujinimo, naudodami *NoUpgrade* argumentą */etc/pacman.conf* byloje:

```
NoUpgrade = boot/grub/menu.lst /etc/vimrc
```

### 3.2.8 Supažindinimas su pacman

Pacman yra geriausias Arch draugas. Labai rekomenduojama išstudijuoti ir išmokti kaip veikia *pacman*.

```
# man pacman
```

## 3.3 Sistemos atnaujinimas

Lieka labai nedaug iki pilno sistemos atnaujinimo. Tiksliau, tai atlikti jau galima ir dabar:

```
# pacman -Syu
```

Arba:

```
# pacman --sync --refresh --sysupgrade
```

Dabar pacman sutikrins visų paketų versijas ir atsiųs, bei įdiegs visus reikiamus paketus. Gali nutikti ir taip, kad pirmiausiai pacman praneš apie savęs pačio naujesnę versiją. Tokiu atveju, atnaujinkite pacman, o paskui pakartokite ankstesnę komandą.

Perkraukite sistemą, jeigu atėjo naujas sisteminis branduolys ( kernel ).

Pacman darbų rezultatas yra saugomas `/var/log/pacman.log`.

### **3.3.1 Arch dabartinės versijos modelis**

Verta prisiminti, jog Arch Linux remiasi dabartinės versijos modeliu. Tai reiškia, jog niekas nebus reikalo diegti sistemą iš naujo. Viskas paremta paprastu principu: kartą įdiegti ir pamiršti. Tiesiog reikia kartais nepamiršti paleisti *pacman -Syu*, sistema atsinaujins ir turėsite naujausius paketus. Taip pat labai patartina perkrauti sistemą keik vieną kartą, kuomet yra išleidžiamas naujas sistemos branduolys ( kernel ).





## skyrius 4

# Naujo vartotojo sukūrimas

Linux yra daugelio naudotojų sistema. Tikrai nereikia naudoti root vartotojo kasdieniams darbams. Tai labai bloga praktika ir tai labai pavojinga. Root vartotojas yra skirtas tik Linux sistemos administravimo darbams. Jis tikrai nėra skirtas programuoti ar rašyti kursinius darbus.

Rekomenduojama sukurti naują, paprastą, ne root tipo vartotoją, pasitelkiant `/usr/sbin/useradd` programą.

```
# useradd -m -g [pradine_grupe] -G [papildomos_grupes] -s  
[prisijungimo_kiautas] [vartotojo_vardas]
```

- **-m** argumentas pasako programai sukurti naują namų direktoriją vartotojui `/home/vartotojo_vardas`. Savo direktorijoje vartotojas turi tiek rašymo, tiek skaitymo teises. Vartotojas savo direktorijoje gali net diegti programas. Taip pat tokioje direktorijoje yra saugomi visos vartotojų konfigūracinės bylos, taip vadinamos 'taškinės bylos' (jų pavadinimas prasideda tašku, kaip pavyzdžiui `.vimrc`). Jos yra 'paslėptos'. Pagal numatytus nustatymus `ls` jų nerodo. Tam, kad jas pamatyti reikia prie komandos pridėti `-a` argumentą. Kuomet kyla konfliktas su vartotojo konfigūracinėmis bylomis ir globaliomis konfigūracinėmis bylomis - vartotojo konfigūracinės bylos lieka dominuojančioje pozicijoje. Dažniausios taškinės bylos, kurios yra keičiamos vartotojo namų direktorijoje yra `.xinitrc` ir `.bashrc`. Sekančios konfigūracinės bylos atsakingos už xinit ir Bash atitinkamai. Šios konfigūracinės bylos leidžia vartotojui keisti langų tvarkyklę, kuri pasirodo po prisijungimo lango, alternatyvius vardus, vartotojo sukurtas komandas, bei aplinkos kintamuosius atitinkamai. Kuomet yra sukuriamas naujas vartotojas, visos jo taškinės bylos, kaip konfigūracijų pagrindas, yra imamos iš `/etc/skel` direktorijos.
- **-g** argumentas nusako prie kokios grupės priskirti vartotoją iškart po prisijungimo prie sistemos. Grupės vardas turi egzistuoti. Jeigu vietoj grupės vardo yra nurodomas grupės numeris, jis turi nurodyti į egzistuojančią grupę. Jeigu parametras nėra nurodomas, `useradd` programos veiksmai priklausys nuo `USERGROUPS_ENAB` kintamojo, kuris yra aprašytas `/etc/login.defs`.

- **-G** argumentas nusako prie kokių grupių priskirti vartotoją. Kiekviena grupė yra pateikiama eilute, grupės atskiriamos kablelių, tarpų būti negali. Pagal numatytus nustatymus, vartotojas priklauso tik tai grupei, kuri yra nusakyta **-g** argumentu.
- **-s** argumentas nusako absoliutų kelią ir pavadinimą numatyto prisijungimo kiauto. Arch Linux init skriptai naudoja Bash kiautą. Po sistemos krovimo, vartotojas bus perkeltas prie jo nurodyto kiauto. Verta įsitikinti, kad vartotojo nurodytas kiautas egzistuoja sistemoje.

Naudingos vartotojų grupės:

- **audio** - garso reikmėms
- **floppy** - diskelių reikmėms
- **lp** - spausdinimo reikmėms
- **optical** - lokalaus optinio diskinio įrenginio reikmėms
- **storage** - laikmenų reikmėms
- **video** - vaizdo, bei sistemos greitinimo reikmėms
- **wheel** - naudojant sudo
- **games** - esant būtinybei turėti rašymo teises žaidimams
- **power** - kompiuterio energijos reikmėms ( išjungimas, perkrovimas )
- **scanner** - skaitytuvo reikmėms

Tipinio vartotojo sukūrimo komanda atrodytų taip:

```
# useradd -m -g users \
    -G audio,lp,optical,storage,video,wheel,games,power,scanner \
    -s /bin/bash avartotojas
```

Toliau, reikia sukurti slaptažodį naujam vartotojui, naudojant *passwd* komandą:

```
# passwd avartotojas
```

Dabar, naujas ne-root vartotojas jau yra pilnai sukurtas. Galite atsijungti iš root vartotojo su *logout* ir prisijungti kaip ne-root vartotojas.

#### 4.0.2 Vartotojo pašalinimas

Jeigu nutinka kažkokia klaida, arba yra būtinybė pakeisti vartotojo prisijungimo vardą, arba dėl kitos priežastis, galima labai lengvai pašalinti sistemos vartotoją su *userdel*:

```
# userdel -r [vartotojo-vardas]
```

- **-r** argumentas pasako programai pašalinti visas su vartotoju susietas bylas ir direktorijas: jo namų direktorija, su visomis bylomis, bei elektroninio pašto vietą.

## 4.1 Sudo diegimas ir konfigūravimas (nebūtina)

Sudo diegimas su pacman yra labai paprastas:

```
# pacman -S sudo
```

Norint pridėti ne-root vartotoją prie sudo vartotojų grupės, pirmiausiai reikia sukonfigūruoti sudo konfigūracinę bylą. Tai galima padaryti su *visudo* komanda, paleidžiant ją su root vartotojo teisėmis.

Pagal numatytus nustatymus, *visudo* komanda naudos *vi* teksto redaktorių. Jeigu nežinote kaip naudotis tokiu teksto redaktoriumi, patartina pasirinkti kitą. Kitą redaktorių galima nustatyti pakoreguojant *EDITOR* aplinkos kintamąjį (pavyzdyje naudosime *nano*):

```
# EDITOR=nano visudo
```

**Pastaba** Reikia įsidėmėti, jog kintamojo deklaravimas bei *visudo* komandos paleidimas vykdomas vienu metu, per vieną eilutę. Jeigu komandas parašyti atskirai, t.y. per dvi eilutes - niekas nesuveiks.

*Visudo* komanda turi atidaryti */etc/sudoers* konfigūracinę bylą. Pastaroji komanda neredaguoja bylos tiesiogiai. Ji perkopijuoja konfigūracinę bylą į laikiną saugyklą (*tmp*) ir atidaro vartotojui redagavimo aplinką (*vi* arba kitą redaktorių, kuris yra nustatomas per *EDITOR* kintamąjį). Tuomet, kai redagavimas yra baigiamas - */etc/sudoers* laikinoji byla yra sutikrinama ir, jeigu konfigūracinis formatas yra teisingas - perrašo pradinę bylą.

**Perspėjimas** Tiesioginis */etc/sudoers* bylos redagavimas yra griežtai nerekomenduojamas. Formato klaidos gali privesti iki to, jog negalima bus niekaip pasiekti root vartotojo.

Anksčiau mes pridėjome mūsų naujai sukurtą ne-root vartotoją prie “wheel” grupės. Dabar tereikia pasakyti *sudo*, kad visi vartotojai, kurie yra “wheel” grupėje gali operuoti sistemoje ir su root teisėmis, kuomet yra vykdoma *sudo* komanda. Tam tereikia atkomentuoti tokią eilutę:

```
%wheel ALL=(ALL) ALL
```

Dabar, jeigu kursite naujus vartotojus, jums tereikės juos pridėti prie “wheel” grupės ir jie galės naudotis root vartotojo teisėmis.



## skyrius 5

# Garsas

Garsui Arch Linux sistemoje apdoroti yra naudojamas “Advanced Linux Sound Architektūra” ( taip pat dar žinomas kaip ALSA ). Jis yra Linux sisteminio branduolio ( kernel ) komponentas, kuris yra skirtas pakeisti originalų “Open Sound System” ( taip pat dar žinoma kaip OSS ). Be garso plokščių atpažinimo, ALSA taip pat atlieka pridėtinės programinės įrangos vaidmenį, t.y. ji suteikia programuotojams pagrindą aukštesnio lygio API, kuris tiesiogiai sąveikauja su sisteminio branduolio ( kernel ) tvarkyklėmis.

**Pastaba** Alsa yra įtraukta į pagrindinį Arch Linux sisteminio branduolio paketą ir udev automatiškai atpažins kompiuterio garso plokštę, bei užkraus reikiamą tvarkyklę/modulį į sisteminį branduolį. Taip pat, garsas sistemoje jau turi veikti, tiesiog pagal numatytus nustatymus visi kanalai yra užslopinti.

Garso kanalams atidaryti reikia *alsamixer* komandos, kuri yra paketo *alsa-utils* dalis. Diegimui įveskite sekančią komandą:

```
# pacman -S alsa-utils
```

Taip pat rekomenduojama diegti *alsa-oss* paketą, kuris suteiks programoms, naudojančioms OSS tiltą prie ALSA:

```
# pacman -S alsa-oss
```

Jeigu ne-root vartotojas dar nėra įtrauktas į *audio* vartotojų grupę, tai galima atlikti su *gpasswd* komanda:

```
# gpasswd -a vartotojo-vardas audio
```

Grupės pridėjimas yra įvykdytas, tačiau dabartinėje vartotojo sesijoje, paprastas vartotojas vis dar nėra prijungtas prie *audio* grupės. Tam, kad nauji nustatymai įsigaliotų, reikia išsiregistruoti iš sistemos ir vėl prisiregistruoti prie sistemos.

Kaip ne-root, paprastas vartotojas įvykdysite *alsamixer*:

```
# alsamixer
```

Norint atidaryti tam tikrą kanalą, pirmą reikia jį įjungti, paspaudžiant “M” raidę. Garso reguliavimas atliekamas su krypties klavišais į viršų ir į apačią. Tam, kad pereiti prie kito kanalo, paspauskite krypties klavišus į dešinę arba į kairę.

Dažniausiai reikia įjungti ir duoti maksimumą “Master” ir jo analogijoms, pavyzdžiui “Master Mono”, “Headphone” ir kiti, kurie yra prieš “PCM”. Su “PCM” galima atlikti pagrindinį sistemos garso reguliavimą.

### 5.0.1 Garso testavimas

Garso nustatymus patikrinti galima su *aplay* komanda:

```
# aplay /usr/share/sounds/alsa/Front_Central.wav
```

Turėtumėt išgirsti malonų moters garsą, kuris sako “Front, center”.

### 5.0.2 Garso nustatymų išsaugojimas

Persijunkite prie root vartotojo ir išsaugokite visus garso nustatymus, naudojant *alsactl* komanda:

```
# alsactl -f /var/lib/alsa/asound.state store
```

Šita komanda sukurs naują bylą, pavadinimu *asound.state*, kurioje bus išsaugoti visi ALSA reikalingi konfigūraciniai duomenys.

Galiausiai, reikia įtraukti *alsa daemon*’ą į *DEAMONS* */etc/rc.conf* masyvą, kad kiekvieną kartą, paleidžiant sistemą, garso nustatymai atsistatytų:

```
DAEMONS=(syslog-ng @network crond alsa)
```

## skyrius 6

# Grafinė vartotojo aplinka

### 6.1 Žingsnis 1: X diegimas

**X** 11 versijos langų sistema ( taip pat žinoma kaip **X11** arba tiesiog **X** ) yra tinklo ir atvaizdavimo prievadas, kuris suteikia langų aplinką taškiniuose vaizduokliuose. Jis turi savyje visus reikiamus įrankius grafinės aplinkos ( GUI ) kūrimui.

**X** suteikia pagrindinę bazę, GUI kūrime: vietą pelytės sąsajai, klaviatūrai ar kitiems įvesties įrenginiams, operuojant su langais darbastalyje. **X** nesuteikia vartotojo aplinkos įgaliojimus kiekvienam langui. Tuo turi pasirūpinti pačios programos. **X** nesuteikia grafinio grožio, kadangi tam reikės dar vieno sluoksnio. **X** yra taip pavadintas, kadangi jis kilo iš **W** langų sistemos, kurią kūrė Stanford'o Universitete.

#### 6.1.1 A: Xorg diegimas

Dabar mes diegsime Xorg, naudodami pacman. Tai yra pirmas žingsnis grafinės aplinkos kūrime:

```
# pacman -S xorg
```

3D palaikymui, reikia diegti mesa:

```
# pacman -S mesa
```

3D įrankiai, tokie kaip *glxgears* arba *glxinfo* yra **mesa-demos** pakete:

```
# pacman -S mesa-demos
```

#### 6.1.2 B: Vaizdo tvarkyklės diegimas

Toliau seka vaizdo tvarkyklės diegimas. Šiame žingsnyje reikalaujama žinoti koks yra kompiuterio lustų rinkinys. Jeigu tai ir nėra žinoma, tai nesunku sužinoti su *lspci* programa:

```
# lspci
```

**Pastaba** `lspci` programa parodo visus lustus, kurie yra prijungti prie kompiuterio per PCI jungtį.

**Pastaba** Jeigu reikiamos tvarkyklės nepavyksta rasti, tuomet iš bėdos galima naudoti ir **vesa** vaizdo tvarkyklę. Tačiau ji siūlo tik lėtą 2D aplinką.

Jeigu reikia diegti atviro kodo tvarkykles, įvykdykite:

```
# pacman -Ss xf86-video | less
```

Arba naudokite šį atrivo kodo tvarkyklių sąrašą

- **xf86-video-apm** - Alliance ProMotion vaizdo tvarkyklė
- **xf86-video-ark** - ark vaizdo tvarkyklė
- **xf86-video-ast** - ASPEED AST vaizdo tvarkyklė
- **xf86-video-ati** - ATI(AMD) radeon vaizdo tvarkyklė
  - **xf86-video-r128** - ATI(AMD) vaizdo tvarkyklė ati Rage128 vaizdo kortai
  - **xf86-video-mach64** - ATI(AMD) vaizdo tvarkyklė mach64 vaizdo kortai
- **xf86-video-chips** - Chips and Technologies vaizdo tvarkyklė
- **xf86-video-cirrus** - Cirrus Logic vaizdo tvarkyklė
- **xf86-video-dummy** - fiktyvi vaizdo tvarkyklė
- **xf86-video-fbdev** - framebuffer vaizdo tvarkyklė
- **xf86-video-glint** - GLINT/Permedia vaizdo tvarkyklė
- **xf86-video-i128** - Number 0 i128 vaizdo tvarkyklė
- **xf86-video-i740** - Intel i740 vaizdo tvarkyklė
- **xf86-video-intel** - Intel i810/i830/i915/945G/G965+ vaizdo tvarkyklė
- **xf86-video-mga** - Mga ( Matrox Graphics Adapter ) vaizdo tvarkyklė
- **xf86-video-neomagic** - neomagic vaizdo tvarkyklė
- **xf86-video-nv** - NVidia nv vaizdo tvarkyklė
- **xf86-video-nouveau** - Open Source 3D acceleration tvarkyklė nVidia vaizdo kortoms
- **xf86-video-openchrome** - VIA/S3G UniChrome, UniChrome Pro ir Chrome9 vaizdo tvarkyklė



- **xf86-video-rendition** - Rendition vaizdo tvarkyklė
- **xf86-video-s3** - S3 vaizdo tvarkyklė
- **xf86-video-s3virge** - S3 Virge vaizdo tvarkyklė
- **xf86-video-savage** - Savage vaizdo tvarkyklė
- **xf86-video-siliconmotion** - Siliconmotion vaizdo tvarkyklė
- **xf86-video-sis** - SiS vaizdo tvarkyklė
- **xf86-video-sisusb** - SiS USB vaizdo tvarkyklė
- **xf86-video-tdfx** - TdFX vaizdo tvarkyklė
- **xf86-video-trident** - Trident vaizdo tvarkyklė
- **xf86-video-tseng** - Tseng vaizdo tvarkyklė
- **xf86-video-unichrome** - VIA S3 Unichrome vaizdo tvarkyklė
- **xf86-video-v4l** - V4l vaizdo tvarkyklė
- **xf86-video-vesa** - Vesa vaizdo tvarkyklė
- **xf86-video-vmware** - Vmware vaizdo tvarkyklė
- **xf86-video-voodoo** - Doovoo vaizdo tvarkyklė
- **xf86-video-xgi** - XGI vaizdo tvarkyklė
- **xf86-video-xgixp** - XGIXP vaizdo tvarkyklė

Norint įdiegti vaizdo tvarkyklę, naudokite pacman, pavyzdžiui:

```
# pacman -S xf86-video-vesa
```

### **NVIDIA vaizdo plokštės**

NVIDIA vartotojai turi tris pasirinkimus ( nepaisant vesa tvarkyklės ):

- Atviro kodo nouveau tvarkyklė, kuri siūlo 2D greitinimą ir eksperimentinį 3D palaikymą, kuris gerai tinka komponavimo ( compositing ) reikmėms, tačiau jis nepalaiko tokių dalykų kaip energijos taupymas.
- Atviro kodo ( tačiau klaidinantis ) nv tvarkyklė, kuri yra labai lėta ir siūlo tik 2D palaikymą.
- Atitinkamas nvidia tvarkyklės, kurios siūlo gerą 3D palaikymą, bei energijos taupymą. Jeigu yra planuose naudoti atitinkamą nvidia tvarkyklę, pirmiausiai reiktų pradėti nuo nouveau tvarkyklės, o tik tuomet persijungti prie nvidia tvarkyklės, kadangi pastaroji veikia iškart, o nvidia reikalaus konfigūracijos ir gali kilti tam tikrų problemų, kurias teks spręsti.

Atviro kodo nouveau yra patenkinama daugumai vartotojų ir yra rekomenduojama:

```
# pacman -S xf86-video-nouveau
```

3D palaikymui ( eksperimentinis ):

```
# pacman -S nouveau-dri
```

Sukurkite naują `/etc/X11/xorg.conf.d/20-nouveau.conf` bylą ir pridėkite sekančią turinį:

```
Section "Device"
    Identifier "n"
    Driver "nouveau"
EndSection
```

Reikia įsitikinti, jog nouveau tvarkyklė bus pakrauta. Xorg dar nėra pakankamai protingas, kad tai atliktų automatiškai.

### ATI vaizdo plokštės

ATI vartotojai turi du tvarkyklių pasirinkimus ( nepaisant vesa tvarkyklės ):

- Atviro kodo **radeon** tvarkyklė, kuri yra paketo **xf86-video-ati** dalis. Ji pilnai palaiko visas ATI kortas iki X1950 ( naujausias R500 lustų rinkinys ). Kortos iki 9200 yra pilnai palaikomos, veikia stabiliai ir suteikia 2D ir 3D greitinimą. Kortos nuo 9500 iki HD4000 suteikia pilną 2D greitinimą, ir iš dalies stabilų 3D greitinimą, tačiau jai trūksta kai kurių specifinių dalykų, kurias siūlo atitinkamų plokščių eilė: energijos taupymas yra vis dar kuriamas ir yra išplėstinėje stadijoje, tačiau jis visiškai atskiras nuo catalyst. HD5000 palaikymas yra dar kūrimo stadijoje. Palaiko KMS ir HDMI su garso išėjimu nuo 2.6.33 branduolio versijos.
- Atitinkama **fglrx** tvarkyklė įtraukta į catalyst paketą yra pasiekama AUR saugykloje. Ji palaiko tik naujausius įrenginius ( HD2xxx ir naujesnius ). Ankščiau paketas buvo įtrauktas į *extra* saugyklą, tačiau 2009 kovą jo palaikymas buvo nutrauktas dėl kokybės ir greičio prastumo.

Atviro kodo tvarkyklė yra rekomenduojamas pasirinkimas:

```
# pacman -S xf86-video-ati
```

### 6.1.3 C: Įvesties ( input ) tvarkyklių diegimas

Udev turi atlikti kompiuterio aparatinės įrangos atpažinimą, o evdev ( `xf86-input-evdev` ) yra modernus, savaiminio atpažinimo įvesties tvarkyklė beveik visiems įrenginiams, tad daugumoje atveju, įvesties tvarkyklių įdiegti nereikia. Šiame žingsnyje, evdev jau yra įdiegtas į kompiuterį kaip Xorg priklausomybė.

Jeigu evdev nepalaiko įrenginio, tuomet reikia diegti tvarkyklę, kuri yra `xorg-input-drivers` grupėje.

Visiškam palaikomų įvesties tvarkyklių sąrašui įvykdykite sekančią komandą:

```
# pacman -Ss xf86-input | less
```

**Pastaba** Tereikia tik `xf86-input-keyboard` arba `xf86-input-mouse`, jeigu planuojate nenaudoti savaiminio atpažinimo.

Nešiojamų kompiuterių savininkams ( arba vartotojams su lietimui jautriais ekranais ) dar reikia diegti *synaptics* paketą, kuris leis **X** sukonfigūruoti jutikliniam kilimėliui arba jutikliniam ekranui:

```
# pacman -S xf86-input-synaptics
```

## 6.2 Žingsnis 2: X konfigūravimas ( nebūtina )

X.org X serveris turi auto konfigūracijos galimybę. Taip pat, X gali funkcionuoti ir visiškai be vartotojo nurodytos konfigūracijos.

Sąrašas konfigūracinių bylų yra `/etc/X11` direktorijoje. Šitame kataloge yra *xorg.conf.d* katalogas, kuriame saugomi visos numatytos konfigūracinės bylos X serveriui. Tai yra Arch Linux distribucijos sukurtos numatytos konfigūracinės bylos, bazė vartotojų redagavimui. Taip pat lieka galimybė tiesiog sukurti paprastą `/etc/X11/xorg.conf`, kuris bus skaitomas paskutinis, po visų **xorg.conf.d** kataloge esančių konfigūracinių bylų.

### 6.2.1 Ne-US klaviatūra

Jeigu naudosite ne standartinę US tipo klaviatūrą, reikia nustatyti klaviatūros išdėstymą `/etc/X11/xorg.conf.d/10-evdev.conf` byloje:

```
Section "InputClass"
    Identifier "evdev keyboard catchall"
    MatchIsKeyboard "on"
    MatchDevicePath "/dev/input/event*"
    Driver "evdev"
    Option "XkbLayout" "lt"
EndSection
```

## 6.3 Žingsnis 3: Bazinės grafinės aplinkos paleidimas

Šitam žingsnyje paaiškinsime kaip paleisti bazinę grafinę aplinką, kuri yra įtrauka į Xorg paketą. Tokia grafinė aplinka naudoja paprasčiausią langų tvarkyklę, *twm*. Numatyta X aplinka yra labai primityvi, tačiau greitu metu diegsime geresnę langų tvarkyklę.

Kadangi Xorg diegimas jau pabaigtas, paprasto vartotojo namų kataloge turi būti konfigūracinė byla `.xinitrc`. Šiame žingsnyje bylą reikia ištrinti. Jeigu bylos nepašalinsite, vietoj tikėtinų langų tvarkyklės bus rodomas tik juodas ekranas, o *Xorg.0.log* nieko nerodys. Paprasčiausiai pašalinkite šitą bylą, kad pasiekti numatytą langų tvarkyklę:

```
# rm ~/.xinitrc
```

### 6.3.1 Pranešimų magistralė

Dbus yra reikalaujamas daugelių programų stabiliam veikimui. Jeigu žinote, jog jis reikalingas nebus - praleiskite sekančią sekciją.

#### Diegimas

Dbus diegiamas su pacman pagalba:

```
# pacman -S dbus
```

#### Deamon paleidimas

Reikia pridėti **dbus** į DEAMONS masyvą */etc/rc.conf* byloje:

```
DEAMONS=(syslog-ng dbus @network crond alsa)
```

Jeigu norite paleisti dbus neperkraunant sistemos:

```
# /etc/rc.d/dbus start
```

### 6.3.2 Xpaleidimas

Galiausiai, galime leisti Xorg:

```
# startx
```

Arba:

```
# xinit -- /usr/bin/X -nolisten tcp
```

Jeigu monitorius rodo tik juodą ekraną, tai dar nėra bėda. Galite persijungti prie kitos konsolės ( pavyzdžiui CTRL+ALT+F2 ) ir prisijungti kaip root.

Galite pabandyti užmušti X serverį su */usr/bin/pkill*:

```
# pkill X
```

**Pastaba** X raidė turi būti didžioji.

Jeigu pkill nesuveiks, tai galima aklai perkrauti kompiuterį:

```
# reboot
```

Arba:

```
# init 6
```

### 6.3.3 Klaidų atveju

Jeigu kyla kažkokia problema, klaidos pranešimo ieškokite `/etc/log/Xorg.0.log` byloje. Ieškokite eilučių, kurios prasideda `EE`, kas reiškia klaidą, bei kurios prasideda `WW`, kas reiškia perspėjimas.

```
# grep EE /var/log/Xorg.0.log
```

Klaidos taip pat rodomos konsolėje, iš kurios buvo leidžiamas `X`.

### 6.3.4 Reikia pagalbos?

Jeigu kyla problemos, kurių išspręsti dar negalite, galite kreiptis į forumus:

- <http://sls.archlinux.lt> - Arch Linux Lietuva forumas
- <http://bbs.archlinux.org> - Oficialus Arch Linux forumas

Taip pat įsitikinkite, jog turite `wgetpaste` paketą:

```
# pacman -S wgetpaste
```

Naudodami įrankį, pateikite forume tokių bylų tekstus:

- `/.xinitrc`
- `/etc/X11/xorg.conf`
- `/var/log/Xorg.0.log`
- `/var/log/Xorg.0.log.old`

`Wgetpaste` naudoti yra labai paprasta:

```
# wgetpaste /absolutus/kelias/iki/bylos
```

## 6.4 Žingsnis 4: Šriftų diegimas

Šiuo metu yra gerai išsaugoti savo laiką ir iškart diegti šriftus. Dejavu yra didelės raiškos ir universalių šriftų rinkinys:

```
# pacman -S ttf-dejavu
```

## 6.5 Žingsnis 5: Grafinės aplinkos pasirinkimas ir diegimas

X langų sistema suteikia pagrindą grafinės vartotojo aplinkos sukūrimui ( GUI ). Langų sistema kontroliuoja vietą, kurioje atsiranda langai, bei jų išvaizdą. Darbalaukio aplinka ( DE ), veikia virš ir su X, suteikdama visiškai funkcinę ir dinamišką GUI. Paprastai DE suteikia langų tvarkyklę, ikonas, programėles, langus, mygtukų juostas, katalogus, darbalaukio fonus, bei pagrindines funkcijas kaip vilkimas.

Kaip alternatyva DE, galite pasirinkti bet kokią langų tvarkyklę ir išsirinkti savo programų rinkinį.

**Pastaba** Daugumos distribucijų nusprendžia už vartotoją kokia darbalaukio aplinka pasirinkti. Arch Linux šitą pasirinkimą palieka vartotojui. Darbalaukio aplinkos ar/ir langų tvarkyklės pasirinkimas yra asmeninis bei skonio reikalas. Pasirinkite geriausiai tinkantį sprendimą pagal esamą poreikį. Rekomenduojama pabandyti visas galimas darbalaukio aplinkas ar/ir langų tvarkykles ir išsirinkti tą, kuri yra arčiausiai širdies.

### 6.5.1 Darbastalio aplinka

### 6.5.2 Langų tvarkyklės

## 6.6 Grafinės aplinkos paleidimo metodai

### 6.6.1 A: Rankinis

Turbūt geriausias sprendimas yra leisti X rankiniu būdu, negu iškart šokti prie darbastalio. Jeigu tai yra nepriimtina - šokite prie sekančio skyriaus “B: Automatinis”. Yra trys galimi rankiniu būdu paleisti X būdai. Pasirinkite tą, kuris yra labiausiai tinkamas:

#### Metodas 1: Specifinės GA komandos

Kai kuriems vartotojams šis metodas yra paprasčiausias, kadangi užtenka tik vienos komandos grafinės aplinkos paleidimui. Paprasčiausiai pasinaudokite komanda, kuri paleidžia tam tikrą darbalaukio aplinką arba langų tvarkyklę. Komandą pradėkite su *xinit* arba *startx* o toliau nurodykite kokią darbo aplinką paleisti:

```
# xinit /usr/bin/gnome-session
```

Arba:

```
# startx /usr/bin/startkde
```

**Pastaba** Sekančios komandos baigia savo vykdymą tik tuomet, kai bus įvykdytas išregistravimas iš darbalaukio aplinkos.

**Pastaba** Naudokite pilną kelią iki darbalaukio aplinkos/langų tvarkyklės. Jei-gu, pavyzdžiui, paleisite gnome kaip *xinit gnome-session*, gali kilti tam tikrų problemų.

Kaip alternatyvą ilgam komandos rašymui, galima sukurti alternatyvų vardą */etc/bash.bashrc* byloje:

```
...
alias launch_gnome="xinit /usr/bin/gnome-session"
alias launch_kde="xinit /usr/bin/startkde"
alias launch_openbox="xinit /usr/bin/openbox-session"
...
```

Tokiu atveju, tereiks surinkti *launch\_gnome* terminale ir pasileis gnome darbastalio aplinka.

Dar galima naudoti prisijungimo vadovą, kuriame galima pasirinkti kokią darbastalio aplinką paleisti. Jei-gu turite gnome, tai jo prisijungimo vadovas yra gdm:

```
# gdm
```

## Metodas 2: Naudojant runlevels

Pagal numatytus nustatymus, Linux yra keli skirtingi runlevels. Arch kraunasi į trečią runlevel. Penktas runlevel naudojamas Linux X serverio krovimui. Pakoreguokite savo */etc/inittab*. Paskutinėje sekcijoje, bylos pabaigoje atkomentuokite atitinkamą eilutę savo darbastalio aplinkos programai. Jei-gu naudojate gnome, tai būtų:

```
# Example lines for starting a login manager
#x:5:respawn:/usr/bin/xdm -nodaemon
x:5:respawn:/usr/sbin/gdm -nodaemon
#x:5:respawn:/usr/bin/kdm -nodaemon
#x:5:respawn:/usr/bin/slim >/dev/null 2>&1
```

Dabar tiesiog paleiskite savo darbastalio programą:

```
# init 5
```

## Metodas 3: Naudojant .xinitrc

Šitas metodas reikalauja daugiausiai konfigūravimo. Pirmiausiai reikia sukonfigūruoti */.xinitrc*. Pagrindinė šios bylos paskirtis yra nurodyti X langų serveriui ką daryti per */usr/bin/startx* arba */usr/bin/xinit* kiekvieno vartotojo asmeni-niam pasirinkimui. Yra begalės konfigūracinių ypatybių, kurias galite įrašyti į savo */.xinitrc*.

*Startx* ar *xinit* paleis X serverį, bei jo klientą. Nuspręsti, kokį klientą paleis-ti, *startx/xinit* pirmiausiai žiūrės į */.xinit*, kuris yra vartotojo namų kataloge.

Jeigu sekančios bylos nėra, komandos žiūri į numatytą konfigūracinę bylą *xinit*, kuri yra */etc/X11/xinit/xinitrc*, kuri sukonfigūruota paleisti TWM langų tvarkyklę.

- Persijunkite į paprastą, ne-root vartotoją:

```
# su - vartotojo-vardas
```

- Kaip šabloną, savo konfigūracijoms, galima naudoti */etc/skel/* kataloge esantį *.xinitrc*. Pavadinimas *skel* yra žodžio *skeleton* trumpinys. Nukopijuokite bylą iš katalogo į savo namų direktoriją:

```
# cp /etc/skel/.xinitrc ~/
```

- Atsidarykite bylą:

```
# nano ~/.xinitrc
```

Ir atkomentuokite eilutę, kuri nusako naudojama darbalaukio aplinkos programą. Jeigu naudosite *gnome*, tai konfigūracinė byla turi atrodyti panašiai:

```
#!/bin/sh
#
# ~/.xinitrc
#
# Executed by startx (run your window manager from here)

exec gnome-session
# exec startkde
# exec startxfce4
# exec wmaker
# exec icewm
# exec blackbox
# exec fluxbox
# exec openbox-session
# ...or the Window Manager of your choice
# exec xterm
```

**Pastaba** Įsitikinkite, jog atkomentavote tik **vieną** eilutę.

Dabar viskas yra paruošta X serverio paleidimui. Paleiskite **X** kaip paprastas, ne-root vartotojas:

```
# startx
```



Arba:

```
# xinit
```

Dabar darbalaukio aplinka turėtų sėkmingai pasirodyti ekrane. Patikrinkite kaip veikia klaviatūra, pelytė.

Jeigu kyla problemų su automatinio įrengimu ( *automount* ), pabandykite savo */.xinitrc* tarp *exec* ir darbalaukio aplinkos komandos įterpti *ck-launch-session*. Pavyzdžiui:

```
exec ck-launch-session gnome-session
```

Tai užtikrins globalių kintamųjų priskyrimą, kai paleisite darbalaukio aplinkos programą. Komanda *Ck-launch-session* išvalo *consolekit* sesiją. *Consolekit* yra programinis pagrindas, kuris yra atsakingas už įvairius vartotojus, sesijas. Jis suteikia programinei įrangai mechanizmą, kuris leidžia programoms pasiekti visus reikiamus kintamuosius. Jis dirba kartu su *dbus* ir kitais įrankiais.

### 6.6.2 B: Automatinis

Daugelis vartotojų gali pasirinkti automatinę grafinės aplinkos pakrovimą. Apie tai dar reiks parašyti, kadangi Pradžiamokslių vadove to nėra..



**skyrius 7**

**Išvados**