

Tartu Ülikool  
Matemaatika-informaatikateaduskond  
Arvutiteaduse instituut

# Programmeerimiskeel NSIS

Erkki Laaneoks

Tartu 2004

# Üldiselt

- Probleemorienteeritud keel
- Protseduurorienteeritud keel
- Scriptimiskeel
- Windows 32 masinatele ainult
- Kompileeritav, mitte intepreteritav



# Kus kasutatakse

- Algselt Nullsoft'i Winamp'i installeerimisprogramm.
- Hiljem võtnud ka teised kasutusele
- Saab kasutada vabavarana

# Tunnusjooned

- Väikesed installatsioonifailid (ca. 34 KB)
- Töötab Win9x/ME/NT/2000/XP/  
2003 versioonidel
- Erinevate keelte toetus
- Laiendatav
- Hetkel beta versiooni tasemel



# Failid ja laiendid

- NSI – NSIS (Nullsoft Scriptable Install System). Põhiline programmifail
- NLF – NSIS Language File. Sisaldab keeleressursse
- NSH – NSIS Header/Include File. Võib sisaldada igasuguseid makrosid, konstante, ka osasid kompilaatori direktiive jms.

The background of the image is a classic marbled paper pattern, often used in book endpapers. It features a complex, organic design of swirling, interlocking shapes in various shades of grey, beige, and off-white, creating a rich, textured appearance.

Nüüd aga keele juurde



# Üldiselt struktuurist

- Alguses asuvad eelkompileerimiskäsud (define, include ja muud spetsiaalkäsud) (võivad olla läbisegi)
- Funktsioonid ja sektsioonid (võivad samuti läbisegi asuda)
- Sektsioonid võivad sisaldada omakorda alamsektsioone
- Realõputunnust ei ole. Võib panna märgi “;”, mis omab kommentaari tähendust

# Muutujatest üldiselt

- Muutujate nimed on tõstutundlikud
- Muutujad on globaalsed (saab kasutada igas funktsioonis ja sektsioonis)
- Stringid märgitakse jutumärkides
- Märk “\$” on stringis eriotstarbeline. “\$” saab stringis “\$\$” kirjutades
- Muutujate tüübimäärangut ei ole.



# Muutujad

- **Enda muutuja defineerimine:**  
\$minuMuutuja
- **Mitmed muutujate nimed sisse ehitatud:**  
*\$INSTDIR (installeerimistee), \$OUTDIR (praegu kehtiv/aktiivne (väljund)kaust), \$CMDLINE (käsurida, millega installeerimine käivitati (parameetritega), \$LANGUAGE (keele kood) jne.*
- **Enda defineeritavad muutujate nimed:**  
\$0-\$9 (soovitavalt globaalne) ja \$R0-\$R9 (soovitavalt lokaalne). (vanast versioonist)
- Enda defineeritud konstandid:  
\$(minu Konstant)

# Automaatselt väärtustatavad

- Väärtustatakse vastavalt keskkonnale automaatselt:

*\$PROGRAMFILES, \$DESKTOP, \$EXEDIR, \$WINDIR, \$SYSDIR, \$TEMP, \$STARTMENU, \$SMPROGRAMS (tavaline koht), \$SMSTARTUP, \$QUICKLAUNCH, \$HWNDPARENT jne.*



# Spetsiaalmuutujad

- Konstantsed, mida saab igal pool kasutada:
  - $\$ \backslash r$  – nn. tagasijooks (*carriage return*) (ASCII 13)
  - $\$ \backslash n$  – nn. revahetus (ASCII 10)
  - $\$ \backslash "$  – jutumärkide kasutamine
- $\$ \{ \text{midagi} \}$  – enda poolt defineeritu kasutamine  
defineeritakse reaga:  
 $! \text{define midagi "midagi suur"}$

# Magasini kasutamine

- Loob võimaluse elemente pinusse panna ja sealt välja võtta.
- Funktsioonid
  - Exch – vahetab esimese elemendi ette antava muutujaga
  - Pop – Annab elemendi magasinist muutujasse
  - Push – lükkab elemendi magasinini



# Operandid

- Kasutatakse koos funktsiooni IntOp. Näide:

IntOp \$2 9 \*\* 3

- +, -, \*, /, % (MOD funktsioon), | (binaarne OR), & (binaarne AND), ^ (binaarne XOR), ~ (bitiviisiline eitus), ! (loogiline eitus), || (loogiline OR), && (loogiline AND), ++, --, \*\* (astendus), E (eksponent), // (ruutjuur), n! (n faktoriaal)

# Stringitöötlus

- Olemas vajalikumad stringide töötlusvahendid:
- **StrCpy** – Stringi kopeerimine. Süntaks:  
**StrCpy** var str [maxpikkus] [alguskoht]  
Nt: StrCpy %0 “Eriti midagi” 6 6  
Kui maxpikkus < 0, siis saab kasutada truncate funktsioonina (lõpust abs(maxpikkus)). Kui alguskoht < 0, siis alustatakse lõpust abs(alguskoht)’ga.



- **StrLen** – String pikkus.
- Süntaks:

**StrLen** var str

**var** on kasutaja poolne muutuja, kuhu salvestatakse pikkus

**str** on string (või ka muutuja), mille pikkust leitakse

- **StrCmp** – stringide võrdlemine
- Süntaks:  
`StrCmp str1 str2 OnVõrdne [MitteVõrdne]`  
OnVõrdne ja MitteVõrdne on märgendid kuhu hüpata.
- Võrdlemine on tõstutundetu.



# Failioperatsioonid

- **FileOpen** – faili avamine.

- Näide:

`FileOpen %4 "$INSTDIR\sula.log" a`

kus %4 on failipide ning a on mood millisena fail avatakse: a – lisamiseks (append); w – kirjutamiseks; r – lugemiseks

# Olulisemad failitöötlusfunktsioonid

- ReadFile – failist andmete lugemine. Süntaks:  
`ReadFile failipide väljundmuutuja [maxpikkus]`
- FileSeek – asukoha nihutamine failis
- FileWrite – faili kirjutamine
- FileClose – faili sulgemine



# Olulisemad failide/kataloogide operatsioonid

- CopyFiles – failide kopeerimine (saab kasutada vaikset kopeerimist või Shelli poolt pakutavat, koos dialoogiga).
- CreateDirectory – Kataloogi loomine
- SearchPath – otsib path'st, kas leidub taoline fail ja kus.
- IfFileExists – kontrollitakse kas fail eksisteerib
- Veel: Delete, Exec, ExecShell, Rename, Rmdir, SetOutPath

# Failide otsimine

- FindFirst – Esimene fail kataloogis. Süntaks:  
`FindFirst` `Pide` failiväljund failimääratlus
- FindNext – Järgmise faili otsimine
- FindClose – Ressurssi (pideme, handle) vabastamine



# Märgendid (Label)

- Märgitakse :MingiLabel
- Märgendile saab minna käsuga *Goto* või siis mõne funktsiooni parameetrina andes
- Märgendid on kehtivad ainult funktsiooni sektsiooni sees
- V. a. märgendid, millel on ees “.” (punkt), mida saab kasutada globaalse märgendina

# Kommentaarisid

- Kommentaarisid algavad semikooloniga või trelliga ja lõpevad rea lõpuga:  
; See koht oli kõige huvitavam siin  
# See on jällegi väike mõttetarkus
- Või siis saab kasutada kuju /\* ... \*/  
/\*  
kommentaarisid on siin, mis võib pikk olla  
ja üle mitme rea \*/



# Enda def. funktsioonid

- Funktsioone saab ka ise defineerida.

Süntaks:

```
Function minuFunktsioon
```

```
    ;mingid käsud
```

```
FunctionEnd
```

- Funktsiooni välja kutsumine:

```
call minuFunktsioon
```

# Sektsioonid (section)

- Section on komponent või osa mida kasutaja saab installeerimisel valida.
- Sektsioonidel on erinevad nimed ja sisaldavad vajalikke käskusid
- Võivad sisaldada alamsektsioone
- Sektsioon “Uninstall” nimeline sektsioon automaatselt täidab eemaldamise käske.



# Callback funktsioonid

- Võimalik sekkuda pakutud sündmuste korral, defineerides vastavanimelise funktsiooni. Näiteks:

Programmi käivitamisel, installeerimise õnnestumisel või ebaõnnestumisel, hiirega sektsioonist üle minnes, Eemaldamist käivitades jne.

# Olulisemad Callback funktsioonid

- `.onGUIInit` – Kasutatakse tavaliselt enne akna kuvamist akna väljanägemis muutmiseks/kujundamiseks jms.
- `.OnInit` – Kui installeerimine on edukalt initsialiseeritud. Näiteks kontrollimaks näiteks, et kas mingi asi on installeeritud jne
- `.onInstFailed` – kui installeerimine kasutaja või mingil muul põhjusel katkestati, siis on mõnikord vaja ajutised failid eemaldada jne



- `.onInstSuccess` – kui installeerimine on edukalt lõpetatud. Vajalik mõnikord readme fail ja/või installeeritud programm ise avada
- `.onVerifyInstDir` – saab kontrollida installeerimisel antava kausta sobivust
- `un.onGUIInit` – eemaldamise programmi käivitamisel välja kutsutav funktsioon
- `un.onUninstFailed` – kui mingil põhjusel eemaldamise programm katkestati
- `un.onUninstSuccess` – kui programmi eemaldamine on edukalt lõpetatud

# Olulisemad Win funktsioonid

- On võimalik kasutada terve rida windows'i poolt pakutavaid funktsioone:
- MessageBox – teadeteakende kuvamiseks  
Süntaks:

MessageBox valikute\_list teatetekst [nuppID märgend]

- Näide:

MessageBox MB\_ICONSTOP|MB\_YESNO “Olete kindel” IDNO LEi IDYES LJah



- SendMessage – Teadete saatmine mingile aknale. Süntaks:

```
SendMessage HWND teateID wparam lparam  
[var(tagastuseks)] [/TIMEOUT=aeg_ms]
```

- Näite:

```
GetDlgItem $R0 $HWNDPARENT 1028
```

```
CreateFont $R1 "Tahoma" 10 700
```

```
SendMessage $R0 ${WM_SETFONT} $R1 0
```

```
SetStaticBkColor $R0 0x00FFFFFF
```

# Kompileerimisaegsed käsud

- Päisfaili (header file) kaasamine:  
`!include "${NSISDIR}\Minu\vajalik.nsh"`
- Eeldefineerida kasutajaliidese elemente defineerides spetsiaalsed konstandid. Nt:  
`!define MUI_ICON (${NSISDIR}\img\iko.ico)`
- `!echo` – väljastab mingi käsu kompileerimise ajal
- `!system` – käivitab mingi käsu ja võib kontrollida antud programmi tagastusväärtust



- `!packhdr` – väljundfaili pakkimise sisse lülitamine
- `!cd` – muudab jooksva kataloog kompilaatori jaoks
- `!include` – lisab mingist failist programmi teksti juurde
- `!ifdef` – kui defineeritud mingi väärtus, siis täidetakse see, muidu mitte. Lisada saab veel lisatingimusi ja neid `|` ja `&` siduda. `!else` defineerimisel täidetakse see kui `ifdef` nurjub. Bloki lõpetab `!endif`

- Näide:

```
!ifdef installerYks
```

```
    BGGradient 000000 308030 008040
```

```
    InstallColors FF8080 000000
```

```
    InstProgressFlags smooth colored
```

```
!else
```

```
    WindowIcon off
```

```
!endif
```

- !undef – selle käsuga dedefineeritakse definatsioon. Näide:

```
!undef installerYks
```



# Makrode kasutamine

- Eeldefineeritud funktsioonid, millele saab ette anda muutujaid jms.
- Kasutamine:

```
Function .onInit
```

```
!insertmacro MUI_INSTALLOPTIONS_EXTRACT "ol.ini"
```

```
!insertmacro MUI_PAGECOMMAND_WELCOME
```

```
!insertmacro MUI_PAGECOMMAND_LICENSE
```

```
FunctionEnd
```

# Makro deklareerimine

- Makro defineerimise näide:

```
!macro makroNimi param1 param2
```

```
# mingid kood
```

```
!macroend
```



# Pluginad/moodulid

- Võimalus kasutada väliseid DLL faile ja sealt importida funktsioone.
- Näidis:  
`MinuDLL::funkike /NOUNLOAD tegutse`  
Minu – DLL faili nimi  
funkike – väljakutsutav funktsioon  
/NOUNLOAD – võti, et DLL jääks alles mällu  
tegutse – funktsioonile ette antav parameeter.

# Faili ol.ini näidis:

[Settings]

NumFields=1

[Field 1]

Type=checkbox

Text=Kas on vaja installida pahna?

Left=0

Right=125

Top=40

Bottom=55

State=1



# DLL'st funk. päring

- Saab ka suvalisest DLL failist mingit funktsiooni kasutada. Näide:

```
System::Call 'kernel32::GetSystemDirectoryA(  
    .r0, *i r1r1) i .r2'
```

Siin kutsutakse kernel32.dll failist (Windows'i kerneli DLL fail) funktsioon nimega GetSystemDirectoryA. Sulgude sees on parameetrid. lõpus on muutuja millesse tagastatakse. t, i – muutujate tüübid; r0, r1, r2 muutujad. \*i – Integer tüüpi viit.

# Akende kujundus

- Võimalik muuta akende kujundust staatiliselt
- Võimalik muuta akende kujundust ja seda otse jooksvalt programmis
- Selleks olemas mitmeid vahendeid



# Kujunduse funktsioonid

- CreateFont – Fondi määramiseks ja muutmiseks
- SetStaticBkColor – Tausta värv
- XPStyle – Win XP stiili aknad (**on** või **off**)
- WindowIcon – Akna ikooni kuvatakse (**on off**)
- InstType – Missugust kujundust kasutatakse

The background of the slide is a light-colored, marbled paper with a complex, organic pattern of grey and white veins. The text is centered on this background.

# Muud enamkasutavad funktsioonid



- CreateShortCut – kiirkorralduse tegemine. Süntaks:
- CreateShortCut link.lnk target.file [parameetrid [icon.fail [icon\_index\_number [start\_options [kiirklahv [kirjeldus]]]]]]
- DetailPrint – installeerimise väljastamise logimise kohta omapoolsete logide kirjutamine (kui tehti midagi programmselt vms)
- Reboot – tehakse masinale taaskäivitus
- Quit – suletakse niipea kui võimalik installeerimisprogramm

# Registrid\*

- Võimalik registritega opereerida
- ReadRegDWORD – täisarvu lugemine:  
`ReadRegDWORD $1 HKLM "Software\MySof" "Ver"`
- DeleteRegKey – kustutab registrivõtme. Kui võtmega /ifempty, siis ainult tühja võtme
- DeleteRegValue – kustutab registrivõtme väärtuse
- WriteRegBin – kirjutab registrisse binaarset teksti



# INI failid

- ReadINIStr – sektsioonist väärtuse lugemine

`ReadINIStr $3 failinimi sektsiooni_nimi sisestuse_nimi`

- DeleteINISec – INI faili sektsiooni kustutamine
- DeleteINIStr – sektsiooni seest valitava väärtuse kustutamine
- WriteINIStr – sektsioonist väärtuse kirjutamine

# Veatöötlus

- Enamuste funktsioonide puhul pannakse vea puhul ülesse nn. vealipuke.
- SetErrors – seatakse vealipuke
- IfErrors – Kontrollitakse vealipukest ja minnakse parameetriga etteantud vastavale märgendile ja nullitakse vealipuke
- ClearErrors – nullitakse vealipuke



# Erinevad keeled

- Saab kasutada suvalisi erinevaid keeli
- Näide:

ComponentText /LANG=\${LANG\_ESTONIAN}

“Komponentide leht“

Kus ComponentText on mingi asi, mille kohta käib.

# Kokkuvõte

- Keel meenutab järgmisi keeli:
  - Basic (nt: funktsioonid ja parameetrid jne)
  - PHP (nt: \$muutuja, meetodid jms kasutamine jne)
  - Assembler (nt: Goto, push, pop jne)
  - C (nt: viidad, funktsioonide nimed, kompilaatori direktiivid jne)
  - Võibolla veel midagi
  - Tegelikult on ta suhteliselt omapärane keel



- Praegu beeta versioonis (läheb vist lõpliku versioonini kõvasti aega)
- Dokumentatsioon puudulik uuemate asjade suhtes. Uued asjad paljuski leida foorumis
- Kasutatakse suhteliselt laialt. Tuntumad neist oleksid: Winamp, DivX, FireBird, FileZilla, CDEx, Apache Tomcat, eMule, SmartFTP, Bloodshed Dev C++, Sun Java Web Start, Etherreal, Cygwin-Lite, LameFE, PHP Editor,



**Täna tähelepanu eest**

**Küsimusi, kommentaar jms?**

**Erkki Laaneoks**