ECS 140A Programming Languages

Fall 2022

Homework 0

About This Assignment

- In ECS 140A, you will be programming in 3 different languages: Go, CLISP, and SWI-Prolog. This assignment asks you to install the required version of these languages, run the provided sample programs in these languages, and submit the resulting output.
- Installing the correct versions of each programming language will ensure that the behavior of your program on your system matches its behavior in our testing environment. You are free to work on any environment as long as the version is correct, but you should test your submission on CSIF before submitting it to Canvas.
- Information about using CSIF computers, such as how to remotely login to CSIF computers from home and how to copy files to/from the CSIF computers using your personal computer, can be found at `http://csifdocs.cs.ucdavis.edu/about-us/csif-general-faq`. Note that VPN is required to access CSIF computers; see this link for more information.
- Post any questions you have on Piazza.
- To use this document, follow the instructions in Section 3 to use CSIF computers, or follow install instructions in Sections 4–6 depending on your operating system. Then follow the instructions in Section 1 to check whether you are using the correct version. Finally, follow the instructions in Section 2 to submit the solution to this assignment.
  - Section 3: CSIF (reference environment)
  - Section 4: Windows
  - Section 5: macOS
  - Section 6: Linux
- Here is a brief summary of the three programming languages

Language version	Go 1.17.4	CLISP 2.49	SWI-Prolog 7.6.4
Print version command	`go version`	`clisp --version`	`swipl --version`
CSIF binary	`/usr/local/go/bin/go`	`/usr/bin/clisp`	`/usr/bin/swipl`

-

1 Version Check

1.1 Go Version Check

- We will be using Go version 1.17.4, which can be downloaded from https://golang.org/dl/

  Run the command `go version` to verify that you have the correct version installed:

  ```
  $ go version
  go version go1.17.4 <other output>
  ```

1.2 CLISP Version Check

- We will be using the GNU CLISP implementation of Common Lisp, version 2.49, which can be installed from https://clisp.sourceforge.io/.

  Use the command `clisp --version` to verify that you have the correct version installed:

  ```
  $ clisp --version
  GNU CLISP 2.49<other output>
  ```

1.3 SWI-Prolog Version Check

- We will be using the SWI-Prolog implementation of the Prolog programming language, version 7.6.4, which can be installed from https://www.swi-prolog.org/download/stable?show=all.

  Use the command `swipl --version` to verify that you have the correct version installed:

  ```
  $ swipl --version
  SWI-Prolog version 7.6.4<other output>
  ```

2 Homework

- Make sure to pass the version check in Section 1 before moving on.
- In this homework, you will run some programs using the languages you have installed, and submit the output of these programs. You do not need to understand how these programs work or modify the programs.
- To complete the assignment (i) download `hw0-handout.zip` from Canvas, (ii) modify the `.txt` files in the `hw0-handout` directory as per the instructions in this document, and (iii) zip the `hw0-handout` directory into `hw0-handout.zip` and upload this `zip` file to Canvas by the due date. **Do not change the file names, create new files, or change the directory structure of `hw0-handout`.**
- **This assignment has to be worked on individually.**
- Begin working on the homework early.

2.1 Go

- From `hw0-handout/go`, run the following command and copy its output to `hello.txt`

```
go run hello.go
```

2.2 CLISP

- From `hw0-handout/lisp`, run the following command and copy its output to `hello.txt`

```
clisp hello.lisp
```

2.3 SWI-Prolog

- From `hw0-handout/prolog`, run the following command and copy its output to `hello.txt`

```
swipl -s hello.plt
```

3 CSIF

- All three languages are installed on CSIF, though some minor changes may be needed on your end. CSIF is also the reference environment we use to grade your homework.
- Information about using CSIF computers, such as how to remotely login to CSIF computers from home and how to copy files to/from the CSIF computers using your personal computer, can be found at http://csifdocs.cs.ucdavis.edu/about-us/csif-general-faq.

3.1 Accessing CSIF Remotely

- Because the UC Davis Campus will be doing remote teaching this Quarter, the Computer Science Instructional Facility, CSIF, will be closed to in-person access until further notice. The link below provides information on how to access the CSIF computers remotely with an ssh terminal connection and a graphical RDP connection.
- https://docs.google.com/document/d/1Zu6FPeiJv2erSm4ldKM1ncPcGL9lmq3KbHWOHzYsB-E/edit?usp=sharing

3.2 Go for CSIF

- Go 1.17.4 is also installed on *all* CSIF machines.

```
$ ssh <kerberos-id>@pc2.cs.ucdavis.edu
<ssh output>
$ go version
go version go1.17.4 linux/amd64
```

3.3 CLISP for CSIF

- CLISP 2.49 is also installed on *all* CSIF machines. For instance,

```
$ ssh <kerberos-id>@pc2.cs.ucdavis.edu
<ssh output>
$ clisp --version
GNU CLISP 2.49<other output>
```

3.4 SWI-Prolog for CSIF

- SWI-Prolog 7.6.4 is also installed on *all* CSIF machines. For instance,

```
$ ssh <kerberos-id>@pc2.cs.ucdavis.edu
<ssh output>
$ swipl --version
SWI-Prolog version 7.6.4<other output>
```

3.5 Working on CSIF computers

- Knowing tools like `vim`, `tmux`, `screen`, and `nano` can help you work on CSIF computers. Here are some tutorials from the Internet that may be helpful.
- Unix / CSIF
  - https://csiflabs.cs.ucdavis.edu/~ssdavis/30/UnixWorkshop.pdf
- `vim`:
  - https://www.howtoforge.com/vim-basics
  - `man vim`
  - `vimtutor`
- `tmux`:
  - https://linuxize.com/post/getting-started-with-tmux/
  - `man tmux`
- `screen`:
  - https://linuxize.com/post/how-to-use-linux-screen/
  - `man screen`
- `nano`:
  - https://www.howtoforge.com/linux-nano-command/
  - `man nano`

3.6 Epilogue

- Congratulations, you have access to all three languages on CSIF computers. Now you can double check the versions of these three languages following Section 1.

4 Windows

- Please follow this chapter if your are using Windows.
- Note that the instructions are not tested on all platforms.

4.1 Go for Windows

1. Go to `https://golang.org/dl/`.
2. Expand "Archived versions".
3. Expand "go1.17.4".
4. Download "go1.17.4.windows-amd64.msi" (for x86 architecture, try "go1.17.4.windows-386.msi")
5. Install following instructions.
6. Open cmd, then type `go version`, and you should see something like "go version go1.17.4 windows/amd64"

4.2 CLISP for Windows

1. Go to `https://clisp.sourceforge.io/`.
2. Click on "Cygwin"
3. "Install Cygwin by running setup-x86_64.exe"
4. During installing, select "Install from Internet"
5. Choose any mirror.
6. In "Cygwin Setup - Select Packages", search for "clisp" and then expand "All" and "Devel", then change column "New" from "Skip" to "2.49-...".
7. After install, double click "Cygwin64 Terminal" on desktop.
8. Type `clisp --version`, then you should see something like "GNU CLISP 2.49+ (2010-07-17)"
9. Type `cd /cygdrive/c/`, then you are in drive C.

4.3 SWI-Prolog for Windows

1. Go to `https://www.swi-prolog.org/download/stable?show=all`
2. Search for "7.6.4", then you can find SWI-Prolog 7.6.4 for Microsoft Windows (64 bit) and SWI-Prolog 7.6.4 for Microsoft Windows (32 bit). Use the one corresponding to your architecture
3. Download executable and install.
4. After install, there should be "SWI-Prolog" in Start menu.
5. You should be able to open a ".pl" file using SWI-Prolog. The title bar should show "SWI-Prolog (..., version 7.6.4)".
6. Open cmd, then type `"C:\Program Files\swipl\bin\swipl.exe" --version`, and you should see something like "SWI-Prolog version 7.6.4 for x64-win64"

4.4 Epilogue

- Congratulations, you have installed all three languages. Now you should double check the versions of these three languages following Section 1.
- Also make sure to have access to these three languages on CSIF computers (see Section 3), the reference environment.

5 macOS

- Please follow this chapter if your are using macOS.
- Note that the instructions are not tested on all platforms.

5.1 Go for macOS

1. Go to https://golang.org/dl/.
2. Expand "Archived versions".
3. Expand "go1.17.4".
4. Download "go1.17.4.darwin-amd64.pkg" and Install.
5. Open terminal, type `go version`, and you should see something like "go version go1.17.4 darwin/amd64"

5.2 CLISP for macOS

1. Go to https://clisp.sourceforge.io/
2. Click on "MacPorts"
3. Go to "installation section", follow instructions.
4. Open a new terminal window, type `port`. You should see something like "MacPorts 2.6.2" (and enter shell mode).
   - If the `port` cannot be found, quit terminal and open a new terminal, or try to use `/opt/local/bin/port` instead.
5. Quit the port shell mode (control + D), then type `sudo port install clisp`
6. Open a terminal and type `clisp --version`, and you should see something like "GNU CLISP 2.49 (2010-07-07) ..."

5.3 SWI-Prolog for macOS

1. Go to https://www.swi-prolog.org/download/stable?show=all
2. Search for "7.6.4", then you can find SWI-Prolog 7.6.4 for MacOSX 10.6 (Snow Leopard) and later on intel
3. Download executable and install.
4. Add the following line to your ~/.zshrc file:

   ```
   export PATH=$PATH:/Applications/SWI-Prolog.app/Contents/MacOS
   ```

5. Open a terminal and type
   `swipl --version`, and you should see something like "SWI-Prolog version 7.6.4 ..."

5.4 Epilogue

- Congratulations, you have installed all three languages. Now you should double check the versions of these three languages following Section 1.
- Also make sure to have access to these three languages on CSIF computers (see Section 3), the reference environment.

6 Linux

- Please follow this chapter if your are using Linux
- Two major distributions of Linux are Debian-based (e.g. Debian, Ubuntu) and RPM-based (e.g. Red Hat Linux, Fedora, CentOS, Oracle Linux). Be sure to know which one you are using.
- Note that the instructions are not tested on all platforms.

6.1 Go for Linux

- You need around 250 MiB free disk space to install
- Go to https://golang.org/dl/
- Expand "Archived versions"
- Expand "go1.17.4"
- Download go1.17.4.linux-amd64.tar.gz
- Extract the archive to `/usr/local`

```
$ ls go1.17.4.linux-amd64.tar.gz
go1.17.4.linux-amd64.tar.gz
$ pwd=$PWD
$ cd /usr/local
$ sudo tar xvf "$pwd/go1.17.4.linux-amd64.tar.gz"
...
$ /usr/local/go/bin/go version
go version go1.17.4 linux/amd64
```

- You should add `/usr/local/go/bin` to your PATH.

```
$ echo 'PATH=/usr/local/go/bin:$PATH' >> ~/.bashrc
$ echo 'PATH=/usr/local/go/bin:$PATH' >> ~/.bash_profile
$ exec bash
$ go version
go version go1.17.4 linux/amd64
```

6.2 CLISP for Linux

- In https://clisp.sourceforge.io/, see "Linux packages" in "Get CLISP".
- For RPM-based Linux, the following command installs the correct version of CLISP:

```
sudo yum install "clisp-2.49.*"
```

- For Debian-based Linux, the following command should be able to install the correct version of CLISP (though not tested):

```
sudo apt-get update
sudo apt-get install clisp
```

- For Debian-based Linux, also try: https://unix.stackexchange.com/a/487556

6.3 SWI-Prolog for Linux

- On Ubuntu ($\geq$ 18.04), you can install SWI-Prolog 7.6.4 using following command:

```
sudo apt-get update
sudo apt-get install swi-prolog
```

- For other distributions it is possible to build from source: https://www.swi-prolog.org/build/unixautotools.txt
- Reference: http://www.codecompiling.net/node/137
- You need around 500 MiB free disk space to install
- For RPM-based Linux, install prerequisites following https://www.swi-prolog.org/build/Redhat.html

```
sudo dnf install \
    cmake \
    ninja-build \
    libunwind \
    gperftools-devel \
    freetype-devel \
    gmp-devel \
    java-1.8.0-openjdk-devel \
    jpackage-utils \
    libICE-devel \
    libjpeg-turbo-devel \
    libSM-devel \
    libX11-devel \
    libXaw-devel \
    libXext-devel \
    libXft-devel \
    libXinerama-devel \
    libXmu-devel \
    libXpm-devel \
    libXrender-devel \
    libXt-devel \
    ncurses-devel \
```

```
    openssl-devel \
    pkgconfig \
    readline-devel \
    libedit-devel \
    unixODBC-devel \
    zlib-devel \
    uuid-devel \
    libarchive-devel \
    libyaml-devel
```

- For Debian-based Linux, install prerequisites following https://www.swi-prolog.org/build/Debian.html

```
sudo apt-get update
sudo apt-get install \
    build-essential cmake ninja-build pkg-config \
    ncurses-dev libreadline-dev libedit-dev \
    libgoogle-perftools-dev \
    libunwind-dev \
    libgmp-dev \
    libssl-dev \
    unixodbc-dev \
    zlib1g-dev libarchive-dev \
    libossp-uuid-dev \
    libxext-dev libice-dev libjpeg-dev libxinerama-dev libxft-dev \
    libxpm-dev libxt-dev \
    libdb-dev \
    libpcre3-dev \
    libyaml-dev \
    default-jdk junit
```

- (It seems that the latest version of SWI-Prolog is using cmake (https://www.swi-prolog.org/build/unix.html), but not for SWI-Prolog 7.6.4)
- Go to https://www.swi-prolog.org/download/stable?show=all
- Search for "7.6.4", then you can find and download SWI-Prolog source for 7.6.4
- Execute the following

```
$ tar xvf swipl-7.6.4.tar.gz
...
$ cd swipl-7.6.4/
$ ./configure
$ make
...
$ sudo make install
```

```
...
$ # Now we need to install plunit, or we cannot run test cases
$ cd packages
$ ./configure
...
$ make
...
$ sudo make install
...
$ swipl --version
SWI-Prolog version 7.6.4 for x86_64-linux
```

6.4 Epilogue

- Congratulations, you have installed all three languages. Now you should double check the versions of these three languages following Section 1.
- Also make sure to have access to these three languages on CSIF computers (see Section 3), the reference environment.