Make newLISP for Android Linux

Download the latest version of the newLISP Android source distribution from the downloads page

This package can also be made using the following command from the main distribution directory:

```
make android_dist
# or
make android_dist_utf8
```

This will make a newlisp-ndk-x.x.x.tgz in the parent directory of the distribution directory where x.x.x is the version number.

It is missing the `semaphore`, which requires libraries that do not exist on Android. It also does not contain the READLINE compile flavor. UTF8 support can be added by making a source package using `make android_dist_utf8`.

In order for `share` to work for more than 4072 bytes a `/data/tmp` directory must be created on the Android device. This directory is also required if more than 4072 data are transferred when using the `spawn` function.

Thanks to *Kanen Flowers* from www.scruffy.tv for doing the adaptation to the Android flavor of Linux and putting together the following instructions:

Steps to compile:

1. Download the android-ndk
2. Make sure your environment works (see above documentation) on your platform
3. Unpack and put newlisp-ndk in the android-ndk directory
4. Compile newlisp

```
kanen (~/Code/android-ndk)$ ./ndk-build -C newlisp-ndk/
make: Entering directory `/Users/kanen/Code/android-ndk/newlisp-ndk'
Compile thumb  : newlisp <= newlisp.c
Compile thumb  : newlisp <= nl-symbol.c
Compile thumb  : newlisp <= nl-math.c
Compile thumb  : newlisp <= nl-list.c
Compile thumb  : newlisp <= nl-liststr.c
Compile thumb  : newlisp <= nl-string.c
Compile thumb  : newlisp <= nl-sock.c
Compile thumb  : newlisp <= nl-import.c
Compile thumb  : newlisp <= nl-xml.c
Compile thumb  : newlisp <= nl-web.c
```

```
Compile thumb  : newlisp <= nl-matrix.c
Compile thumb  : newlisp <= nl-debug.c
Compile thumb  : newlisp <= pcre.c
Compile thumb  : newlisp <= nl-filesys.c
Executable     : newlisp
Install        : newlisp => libs/armeabi/newlisp
make: Leaving directory `/Users/kanen/Code/android-ndk/newlisp-ndk'
```

The binary gets put into `libs/armeabi/newlisp`. From there, you just have to build an Android emulator and you can copy newLISP over to it.

For that, you need the android-sdk.

First, list the targets and find the armeabi version you want to use. I use Target 4, which isn't the most recent, but is the most widely deployed, at API level 15.

```
kanen (~/Code/android-sdk/tools)$ ./android list targets
Available Android targets:
id: 4 or "Google Inc.:Google APIs:15"
     Name: Google APIs
     Type: Add-On
     Vendor: Google Inc.
     Revision: 2
     Description: Android + Google APIs
     Based on Android 4.0.3 (API level 15)
     ABIs : armeabi-v7a
```

Now, generate an emulator from that target platform:

```
kanen (~/Code/android-sdk/tools)$ ./android create avd -n MyEmulator -t 4
```

Then, run the emulator:

```
kanen (~/Code/android-sdk/tools)$ ./emulator -avd MyEmulator &
```

Attach a shell to the emulator:

```
kanen (~/Code/android-sdk/platform-tools)$ ./adb shell
```

Make sure everything's ok and create a directory for newLisp:

```
# mkdir /data/nl
# chmod 777 /data/nl
# exit
```

Copy newLisp to the emulator:

```
./adb push ~/Code/android-ndk/newlisp-ndk/libs/armeabi/newlisp /data/nl
```

Go back into the shell and run newLisp:

```
kanen (~/Code/android-sdk/platform-tools)$ ./adb shell
# cd /data
# cd nl
# ls
newlisp
# ./newlisp
newLISP v.10.4.5 on Linux IPv4/6, execute 'newlisp -h' for more info.

>
```

+++