iPhone Debugger - Documentation

Project
iPhone Reversing Toolkit

Author
Nicolas Economou (neconomou)

Description

This tool is useful for debugging running (or newly created) native processes inside iPhone.

The development of this debugger is based on a previous Windows debugging tool (`nicodbg`, unreleased) and iPhone's debug API is inspired on the Patrick Walton's (with hdm's updates) weasel debugger.

It was developed on C++ and runs in native code inside iPhone. It has a console interface, similar to that of `ntsd.exe`, a debugger included in all Windows versions.

The design divides the tool into two parts, the interface and the C++ class for debugging, this enables the possibility of making another debugging tools with different interfaces. This design is simple and the debugger could be easily ported to another platforms.

Setup

Copy the executable `iphonedbg` to your preferred folder inside iPhone using OpenSSH Secure Copy (`scp` or WinSCP) or be any means possible.

Its usage is very simple, you've only need to be familiarized with console applications.

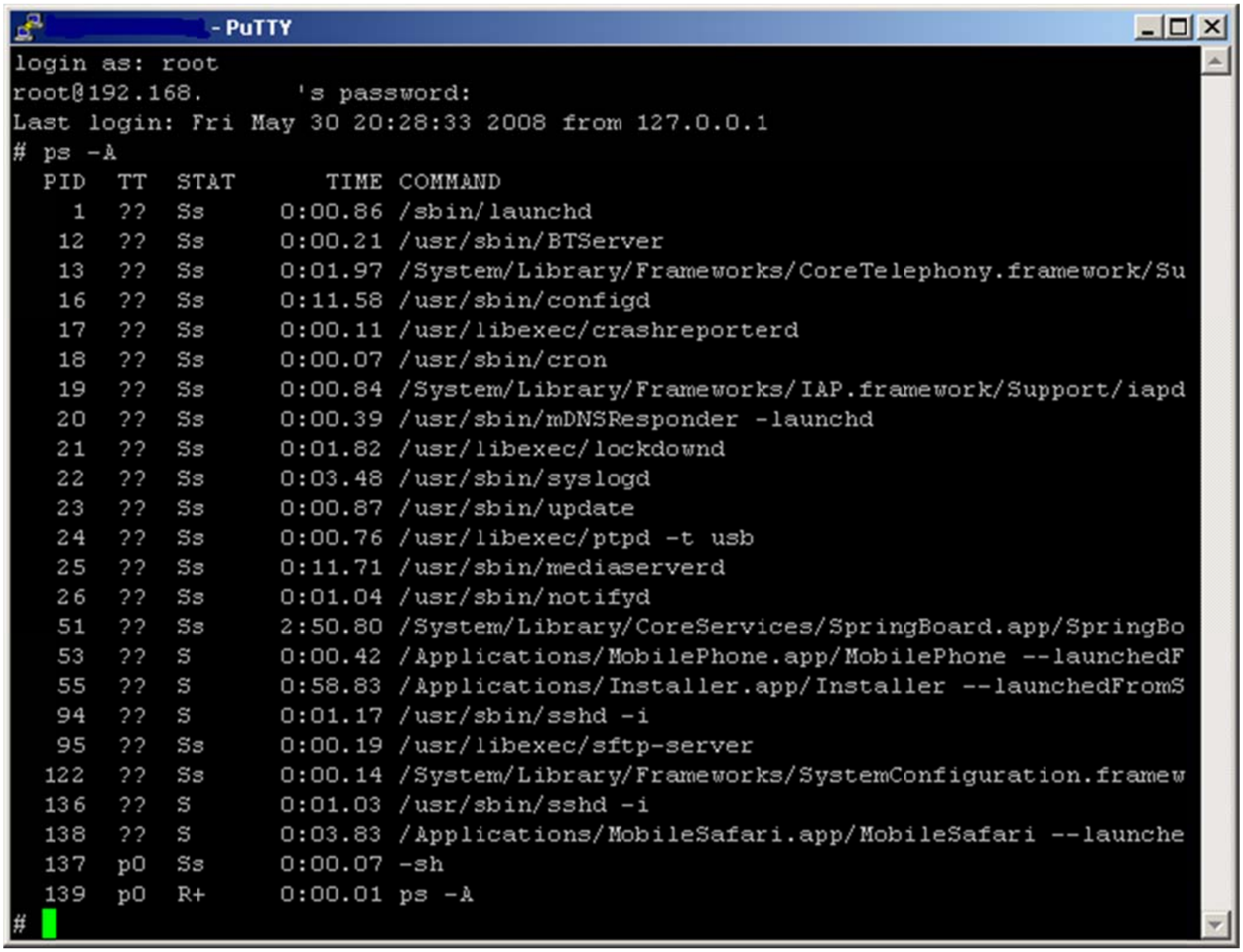
The working command and arguments follow:

```
iphonedbg [-e executable [arguments...]|-p pid executable]
```

The debugger can attach to a running process or start the process from scratch.

Real Life Example

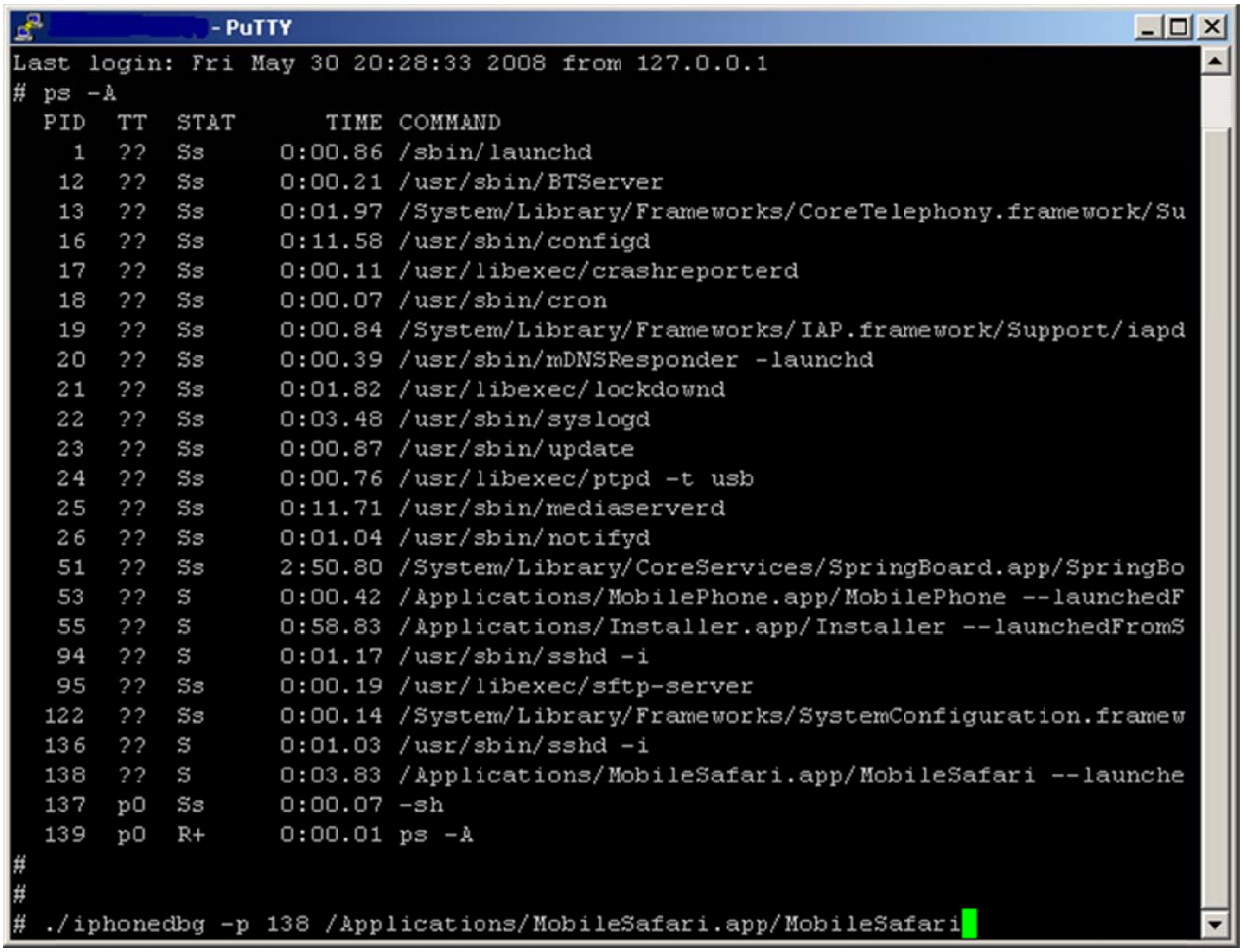
- We run Safari web browser in the iPhone and we list all the active processes from a console conected through `ssh`.



[redacted] - PuTTY

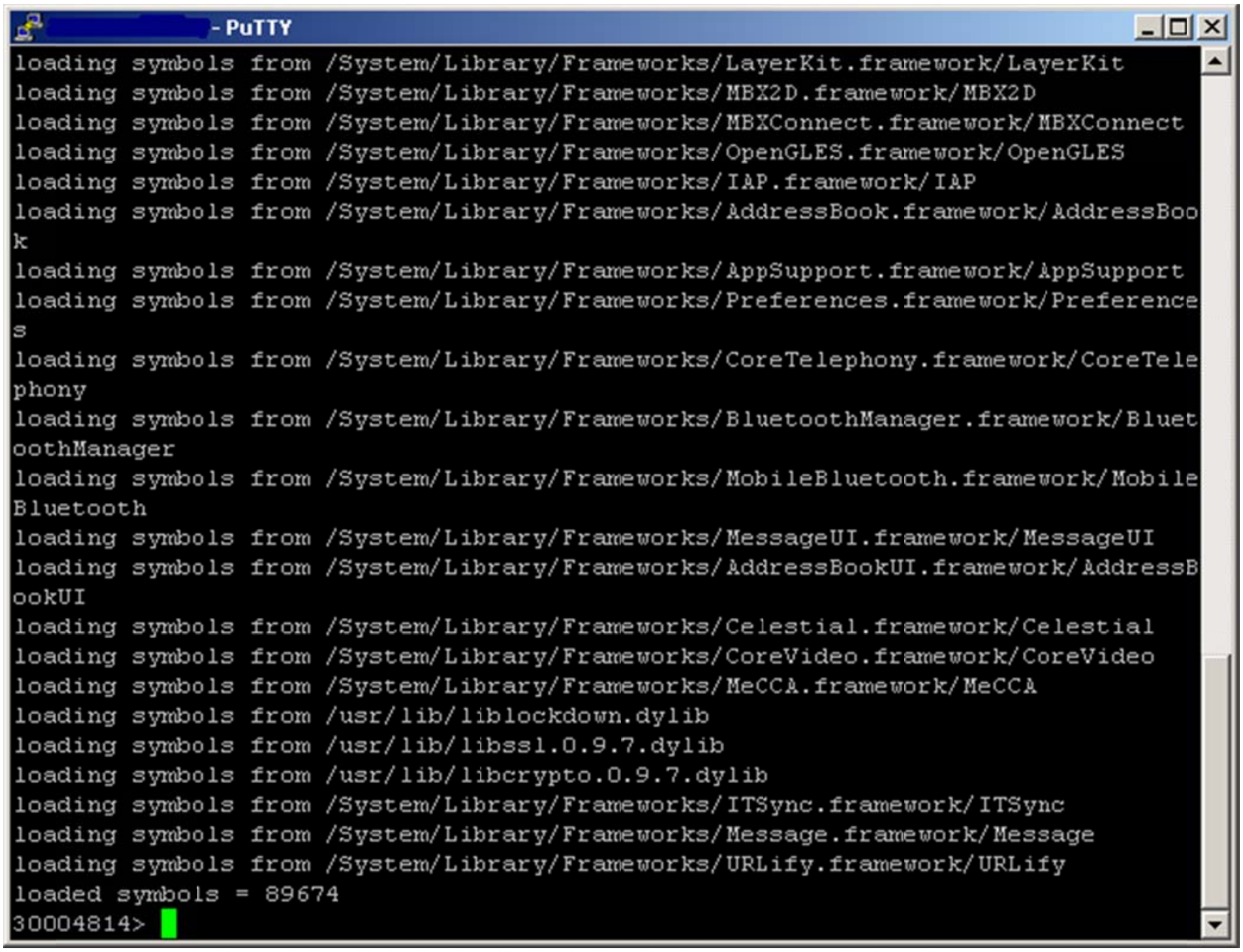
```
login as: root
root@192.168.      's password:
Last login: Fri May 30 20:28:33 2008 from 127.0.0.1
# ps -A
  PID  TT  STAT      TIME COMMAND
    1  ??  Ss     0:00.86 /sbin/launchd
   12  ??  Ss     0:00.21 /usr/sbin/BTServer
   13  ??  Ss     0:01.97 /System/Library/Frameworks/CoreTelephony.framework/Su
   16  ??  Ss     0:11.58 /usr/sbin/configd
   17  ??  Ss     0:00.11 /usr/libexec/crashreporterd
   18  ??  Ss     0:00.07 /usr/sbin/cron
   19  ??  Ss     0:00.84 /System/Library/Frameworks/IAP.framework/Support/iapd
   20  ??  Ss     0:00.39 /usr/sbin/mDNSResponder -launchd
   21  ??  Ss     0:01.82 /usr/libexec/lockdownd
   22  ??  Ss     0:03.48 /usr/sbin/syslogd
   23  ??  Ss     0:00.87 /usr/sbin/update
   24  ??  Ss     0:00.76 /usr/libexec/ptpd -t usb
   25  ??  Ss     0:11.71 /usr/sbin/mediaserverd
   26  ??  Ss     0:01.04 /usr/sbin/notifyd
   51  ??  Ss     2:50.80 /System/Library/CoreServices/SpringBoard.app/SpringBo
   53  ??  S      0:00.42 /Applications/MobilePhone.app/MobilePhone --launchedF
   55  ??  S      0:58.83 /Applications/Installer.app/Installer --launchedFromS
   94  ??  S      0:01.17 /usr/sbin/sshd -i
   95  ??  Ss     0:00.19 /usr/libexec/sftp-server
  122  ??  Ss     0:00.14 /System/Library/Frameworks/SystemConfiguration.framew
  136  ??  S      0:01.03 /usr/sbin/sshd -i
  138  ??  S      0:03.83 /Applications/MobileSafari.app/MobileSafari --launche
  137  p0  Ss     0:00.07 -sh
  139  p0  R+     0:00.01 ps -A
# 
```

- We write the command to attach the debugger to the Safari process.



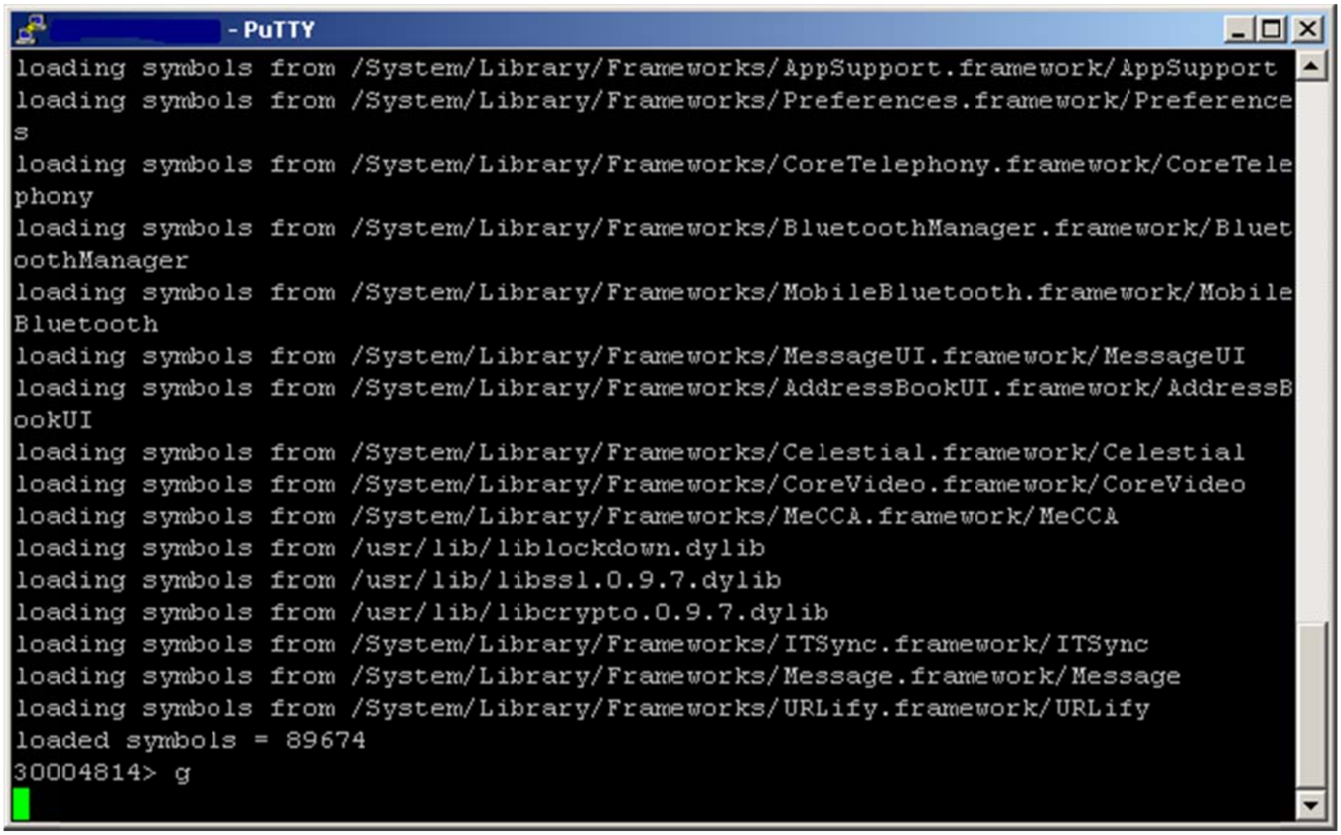
```
Last login: Fri May 30 20:28:33 2008 from 127.0.0.1
# ps -A
  PID  TT  STAT      TIME COMMAND
    1  ??  Ss     0:00.86 /sbin/launchd
   12  ??  Ss     0:00.21 /usr/sbin/BTServer
   13  ??  Ss     0:01.97 /System/Library/Frameworks/CoreTelephony.framework/Su
   16  ??  Ss     0:11.58 /usr/sbin/configd
   17  ??  Ss     0:00.11 /usr/libexec/crashreporterd
   18  ??  Ss     0:00.07 /usr/sbin/cron
   19  ??  Ss     0:00.84 /System/Library/Frameworks/IAP.framework/Support/iapd
   20  ??  Ss     0:00.39 /usr/sbin/mDNSResponder -launchd
   21  ??  Ss     0:01.82 /usr/libexec/lockdownd
   22  ??  Ss     0:03.48 /usr/sbin/syslogd
   23  ??  Ss     0:00.87 /usr/sbin/update
   24  ??  Ss     0:00.76 /usr/libexec/ptpd -t usb
   25  ??  Ss     0:11.71 /usr/sbin/mediaserverd
   26  ??  Ss     0:01.04 /usr/sbin/notifyd
   51  ??  Ss     2:50.80 /System/Library/CoreServices/SpringBoard.app/SpringBo
   53  ??  S      0:00.42 /Applications/MobilePhone.app/MobilePhone --launchedF
   55  ??  S      0:58.83 /Applications/Installer.app/Installer --launchedFromS
   94  ??  S      0:01.17 /usr/sbin/sshd -i
   95  ??  Ss     0:00.19 /usr/libexec/sftp-server
  122  ??  Ss     0:00.14 /System/Library/Frameworks/SystemConfiguration.framew
  136  ??  S      0:01.03 /usr/sbin/sshd -i
  138  ??  S      0:03.83 /Applications/MobileSafari.app/MobileSafari --launche
  137  p0  Ss     0:00.07 -sh
  139  p0  R+     0:00.01 ps -A
#
#
# ./iphonedbg -p 138 /Applications/MobileSafari.app/MobileSafari
```

- Once we are attached to the process, the debugger loads all the process symbols and then waits for user commands.



```
loading symbols from /System/Library/Frameworks/LayerKit.framework/LayerKit
loading symbols from /System/Library/Frameworks/MBX2D.framework/MBX2D
loading symbols from /System/Library/Frameworks/MBXConnect.framework/MBXConnect
loading symbols from /System/Library/Frameworks/OpenGLES.framework/OpenGLES
loading symbols from /System/Library/Frameworks/IAP.framework/IAP
loading symbols from /System/Library/Frameworks/AddressBook.framework/AddressBoo
k
loading symbols from /System/Library/Frameworks/AppSupport.framework/AppSupport
loading symbols from /System/Library/Frameworks/Preferences.framework/Preference
s
loading symbols from /System/Library/Frameworks/CoreTelephony.framework/CoreTele
phony
loading symbols from /System/Library/Frameworks/BluetoothManager.framework/Bluet
oothManager
loading symbols from /System/Library/Frameworks/MobileBluetooth.framework/Mobile
Bluetooth
loading symbols from /System/Library/Frameworks/MessageUI.framework/MessageUI
loading symbols from /System/Library/Frameworks/AddressBookUI.framework/AddressB
ookUI
loading symbols from /System/Library/Frameworks/Celestial.framework/Celestial
loading symbols from /System/Library/Frameworks/CoreVideo.framework/CoreVideo
loading symbols from /System/Library/Frameworks/MeCCA.framework/MeCCA
loading symbols from /usr/lib/liblockdown.dylib
loading symbols from /usr/lib/libssl.0.9.7.dylib
loading symbols from /usr/lib/libcrypto.0.9.7.dylib
loading symbols from /System/Library/Frameworks/ITSync.framework/ITSync
loading symbols from /System/Library/Frameworks/Message.framework/Message
loading symbols from /System/Library/Frameworks/URLify.framework/URLify
loaded symbols = 89674
30004814>
```

- We execute command g (go) and the process continues its execution waiting for some event or exception.



```
loading symbols from /System/Library/Frameworks/AppSupport.framework/AppSupport
loading symbols from /System/Library/Frameworks/Preferences.framework/Preference
s
loading symbols from /System/Library/Frameworks/CoreTelephony.framework/CoreTele
phony
loading symbols from /System/Library/Frameworks/BluetoothManager.framework/Bluet
oothManager
loading symbols from /System/Library/Frameworks/MobileBluetooth.framework/Mobile
Bluetooth
loading symbols from /System/Library/Frameworks/MessageUI.framework/MessageUI
loading symbols from /System/Library/Frameworks/AddressBookUI.framework/AddressB
ookUI
loading symbols from /System/Library/Frameworks/Celestial.framework/Celestial
loading symbols from /System/Library/Frameworks/CoreVideo.framework/CoreVideo
loading symbols from /System/Library/Frameworks/MeCCA.framework/MeCCA
loading symbols from /usr/lib/liblockdown.dylib
loading symbols from /usr/lib/libssl.0.9.7.dylib
loading symbols from /usr/lib/libcrypto.0.9.7.dylib
loading symbols from /System/Library/Frameworks/ITSync.framework/ITSync
loading symbols from /System/Library/Frameworks/Message.framework/Message
loading symbols from /System/Library/Frameworks/URLify.framework/URLify
loaded symbols = 89674
30004814> g
```

If after consulting a web page the Safari process crashes the debugger will inform it.

Help

Help is incorporated to the debugger accessing with the command h, the result is the following:

```
h                          -help
v                          -version
q                          -quit program
r [reg[=expression]]       -print or set registers
g [expression]             -run
t [value]                  -trace execution n times
p                          -trace execution not entering to calls
u [expression]             -print code
db expression              -read byte format memory
dd expression              -read dword format memory
eb expression b1 b2 ...    -write byte format memory
bp expression [condition] -set breakpoint [reg{<|<=|==|>=|>|<>}value]
bc expression              -clear breakpoint
m                          -show memory map
s addr1 addr2 b1 b2 ...    -search from addr1 to addr2
f addr1 addr2 byte         -fill from addr1 to addr2 with byte value
~                          -threads list
~<0..n>r                   -show register values from the thread number
'enter'                    -repeat last command

note: * to set breakpoints in thumb mode write address+1.
      * to execute many cmds in a line use ';'.
      * to execute many times a line: ex. 'repeat 3:r;g;'.
```

Compiling From Scratch

There are two possibilites:

- Downloading the known toolchain contained on a VMWare Linux image, iPhoneToolChainV2:
    - Copy `iphonedbg-v?.?.zip` to the VMWare Linux image and decompress it.
    - Run the following commands to compile the debugger:

```
          iPhoneToolchainV2:~/iphonedbg-v1.1# arm-apple-darwin-gcc -c disasm.c
          iPhoneToolchainV2:~/iphonedbg-v1.1# arm-apple-darwin-g++ -L/usr/local/lib -o
iphonedbg iphonedbg.cpp disasm.o
```

- Download the iPhone SDK for Mac OS X from Apple or from iPhoneFix.de (not tested yet).

iPhone Crashes

When an application crashes inside iPhone a `.plist` file is generated on directory `/private/var/logs/CrashReporter`. This is basically an XML file with the state of the register, thread and the exception type generated. If it is a kernel crash is written at`/private/var/logs/CrashReporter/Panics`.