CSEN 383 – Assignment 1
Winter 2024

Chaitra Boggaram (W1651213)

Working on MacOS

1. Be able to have reliable Linux environment

Running on an M2 MacBook

```
ipadhu@Padhus-MBP Downloads % sw_vers
ProductName:            macOS
ProductVersion:         14.2
BuildVersion:           23C64
ipadhu@Padhus-MBP Downloads %
```

2. Compile and run the C program (forktest.c) on a Linux system and to provide a screenshot of the compilation and execution.

gcc forktest.c -o forktest
forktest

```
ipadhu@Padhus-MBP Downloads % gcc forktest.c -o forktest
ipadhu@Padhus-MBP Downloads % ./forktest
Parent: Process started
Parent: Forking a child.
Parent: Wait for child to complete.
Child: Process started.
Child: Start 10 second idle: 10  9  8  7  6  5  4  3  2  1  0 done!
Child: Terminating.
Parent: Terminating.
ipadhu@Padhus-MBP Downloads %
```

3. Read the gcc man page

I am running man clang because of the below explanations and gcc -v shows the clang version as provided in the below image.

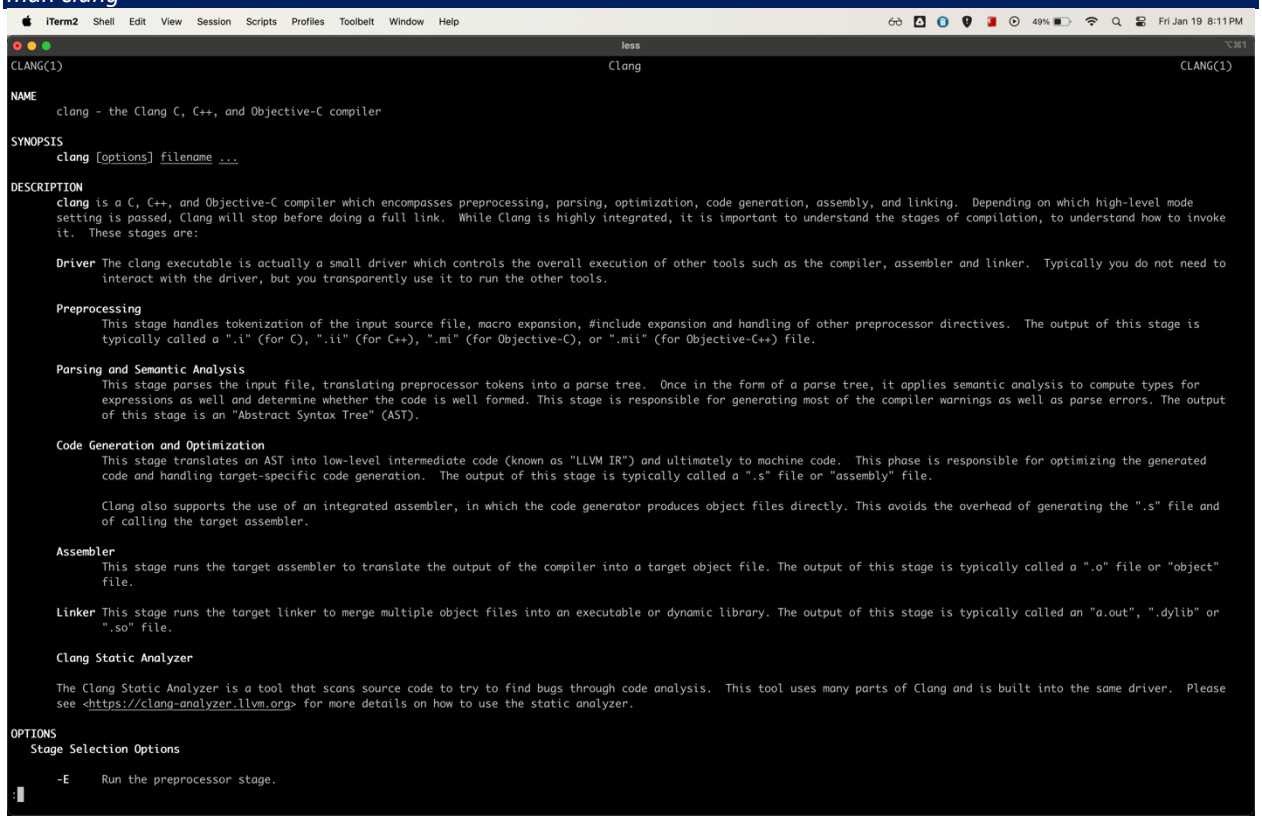
gcc -v

```
ipadhu@Padhus-MBP Downloads % gcc -v
Apple clang version 15.0.0 (clang-1500.0.40.1)
Target: arm64-apple-darwin23.2.0
Thread model: posix
InstalledDir: /Applications/Xcode.app/Contents/Developer/Toolchains/XcodeDefault.xctoolchain/usr/bin
ipadhu@Padhus-MBP Downloads %
```

Apple uses Clang instead of GCC for a few reasons:
Speed and memory: Clang is faster and uses less memory than GCC.
Diagnostics: Clang provides clearer and more concise error and warning messages than GCC.
Language compliance: Clang has better language compliance than GCC, which improves compatibility with newer standards and languages.
Integration: GCC's large code doesn't integrate well with Apple's IDE.
Optimization and reliability: Clang is better than GCC for optimization and reliability because of LLVM.
Clang can replace GCC in many cases without affecting the toolchain. It supports most of the commonly used GCC options. However, for some languages, such as Ada, LLVM is still dependent on GCC or another compiler front end.
Apple doesn't ship a compiler with macOS or iOS, and doesn't want any user-compiled code to run. However, Xcode command-line developer tools include Clang.

Reference:
https://www.incredibuild.com/integrations/clang#:~:text=The%20original%20contributor%20of%20Clang,IDE%20(Integrated%20development%20environment)

man clang



Working on NoMachine (Linux machine from SCU)

1. Be able to have reliable Linux environment

Running on an Linux machine

```
[cboggaram@linux10605 Downloads]$ lsb_release -a
LSB Version:    :core-4.1-ia32:core-4.1-noarch
Distributor ID: CentOS
Description:    CentOS Linux release 7.9.2009 (Core)
Release:        7.9.2009
Codename:       Core
[cboggaram@linux10605 Downloads]$
```

2. Compile and run the C program (forktest.c) on a Linux system and to provide a screenshot of the compilation and execution.

gcc forktest.c -o forktest
forktest

```
[cboggaram@linux10605 Downloads]$ gcc forktest.c -o forktest
[cboggaram@linux10605 Downloads]$ ./forktest
Parent: Process started
Parent: Forking a child.
Parent: Wait for child to complete.
Child: Process started.
Child: Start 10 second idle: 10  9  8  7  6  5  4  3  2  1  0 done!
Child: Terminating.
Parent: Terminating.
[cboggaram@linux10605 Downloads]$ man gcc
[cboggaram@linux10605 Downloads]$
```

3. Read the gcc man page

man gcc