بسم الله الرحمن الرحيم

GCC & Make

Compiling, Linking and Building C/C++ Applications

mahsan.co

سید سروش
حسین‌علی‌پور

تاریخچه

Richard Stallman

The founder of the GNU Project

GCC

- GNU C Compiler
- GNU Compiler Collection

Support Many languages

- C gcc
- C++ g++
- Fortran gfortran
- Go gccgo

GNU Toolchain

- gcc
- GNU Make
- GNU Binutils
- GNU Debugger GDB
- GNU Autotools: A Build system (Autoconf, Autoheader, Automake, Libtool)
- GNU Bison

نسخه زبان

GCC Versions

- GCC version ۱ ۱۹۸۷
- GCC version ۲ ۱۹۹۲: supports C++
- GCC ۱۰.۲
- GCC ۹.۳
- Development GCC ۱۰.۰

C++ Standard Support

- C++۹۸ –std=c++۹۸ or -std=gnu++۹۸
- C++۱۱ –std=c++۱۱ or –std=gnu++۱۱
- C++۱۴ –std=c++۱۴ or –std=gnu++۱۴
- C++۱۷ –std=c++۱۷ or –std=gnu++۱۷
- C++۲۰ –std=c++۲a or –std=gnu++۲a

sudo apt install g++

g++ --version

```
g++ (Ubuntu 9.3.0-10ubuntu2) 9.3.0
Copyright (C) 2019 Free Software Foundation, Inc.
This is free software; see the source for copying conditions. There is NO
warranty; not even for MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.
```

man g++ | col –b > gcc.txt

whereis g++

شروع کنیم

test > inter-node > reliable-multicast > C++ hello.cpp > ...

```cpp
// ITNOA

#include <cstdio>
#include <iostream>

using namespace std;

int main()
{
    cout << "SALAM" << endl;
    return EXIT_SUCCESS;
}
```

- g++ hello.cpp
- chmod a+x a.out
- ./a.out
- g++ -o <output name> hello.cpp

کمی جلوتر

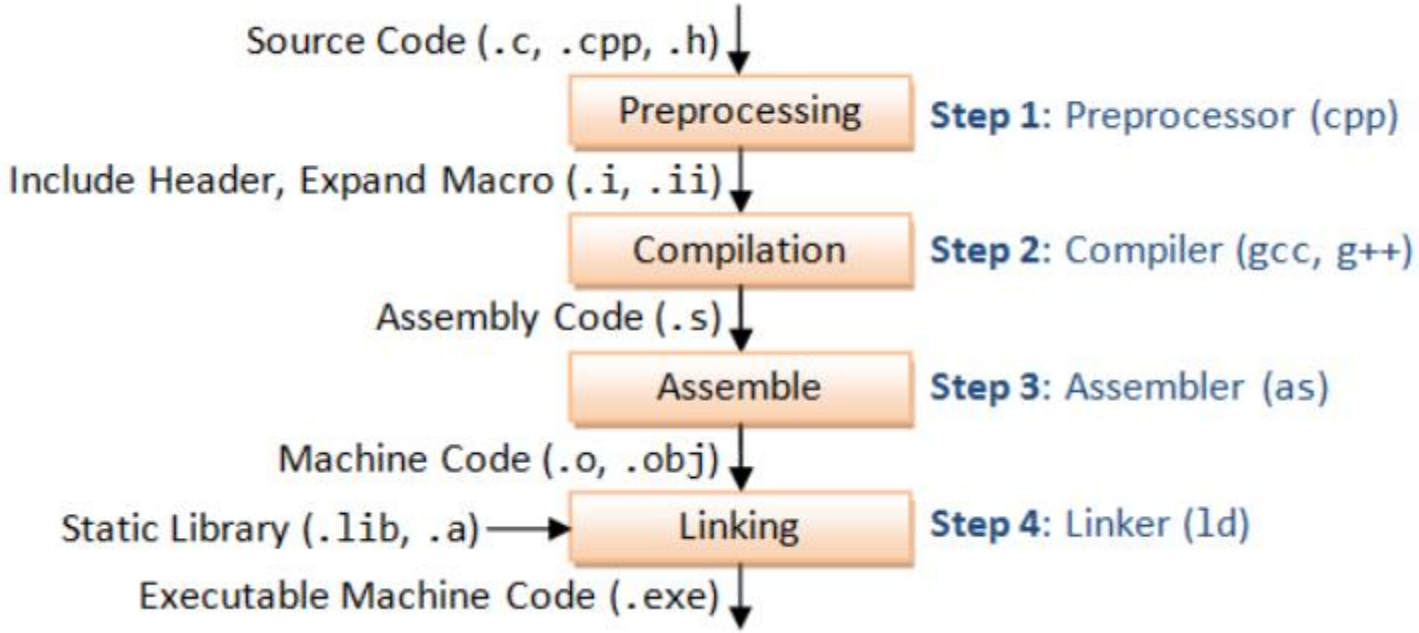
```
soroosh@ssoroosh-pc:/mnt/c/Users/ssoroosh/temp$ ls
Hello.cpp
soroosh@ssoroosh-pc:/mnt/c/Users/ssoroosh/temp$ g++ -c -Wall -g -o Hello.o Hello.cpp
soroosh@ssoroosh-pc:/mnt/c/Users/ssoroosh/temp$ ls
Hello.cpp  Hello.o
soroosh@ssoroosh-pc:/mnt/c/Users/ssoroosh/temp$ g++ -g -o Hello.exe Hello.o
soroosh@ssoroosh-pc:/mnt/c/Users/ssoroosh/temp$ ls
Hello.cpp  Hello.exe  Hello.o
soroosh@ssoroosh-pc:/mnt/c/Users/ssoroosh/temp$
```

Compile and Link Separately

```
$ g++ -Wall -g -o Hello Hello.cpp
```

More GCC Compiler Options

- -o: Specifies the output
- -Wall: prints all warning
- -g: generates additional symbolic


Source Code (.c, .cpp, .h)
Preprocessing
Step 1: Preprocessor (cpp)
Include Header, Expand Macro (.i, .ii)
Compilation
Step 2: Compiler (gcc, g++)
Assembly Code (.s)
Assemble
Step 3: Assembler (as)
Machine Code (.o, .obj)
Static Library (.lib, .a)
Linking
Step 4: Linker (ld)
Executable Machine Code (.exe)

1. Pre-processing: via the GNU C Preprocessor (`cpp.exe`), which includes the headers (`#include`) and expands the macros (`#define`).

```
> cpp hello.c > hello.i
```

   The resultant intermediate file "`hello.i`" contains the expanded source code.

2. Compilation: The compiler compiles the pre-processed source code into assembly code for a specific processor.

```
> gcc -S hello.i
```

   The `-S` option specifies to produce assembly code, instead of object code. The resultant assembly file is "`hello.s`".

3. Assembly: The assembler (`as.exe`) converts the assembly code into machine code in the object file "`hello.o`".

```
> as -o hello.o hello.s
```

4. Linker: Finally, the linker (`ld.exe`) links the object code with the library code to produce an executable file "`hello.exe`".

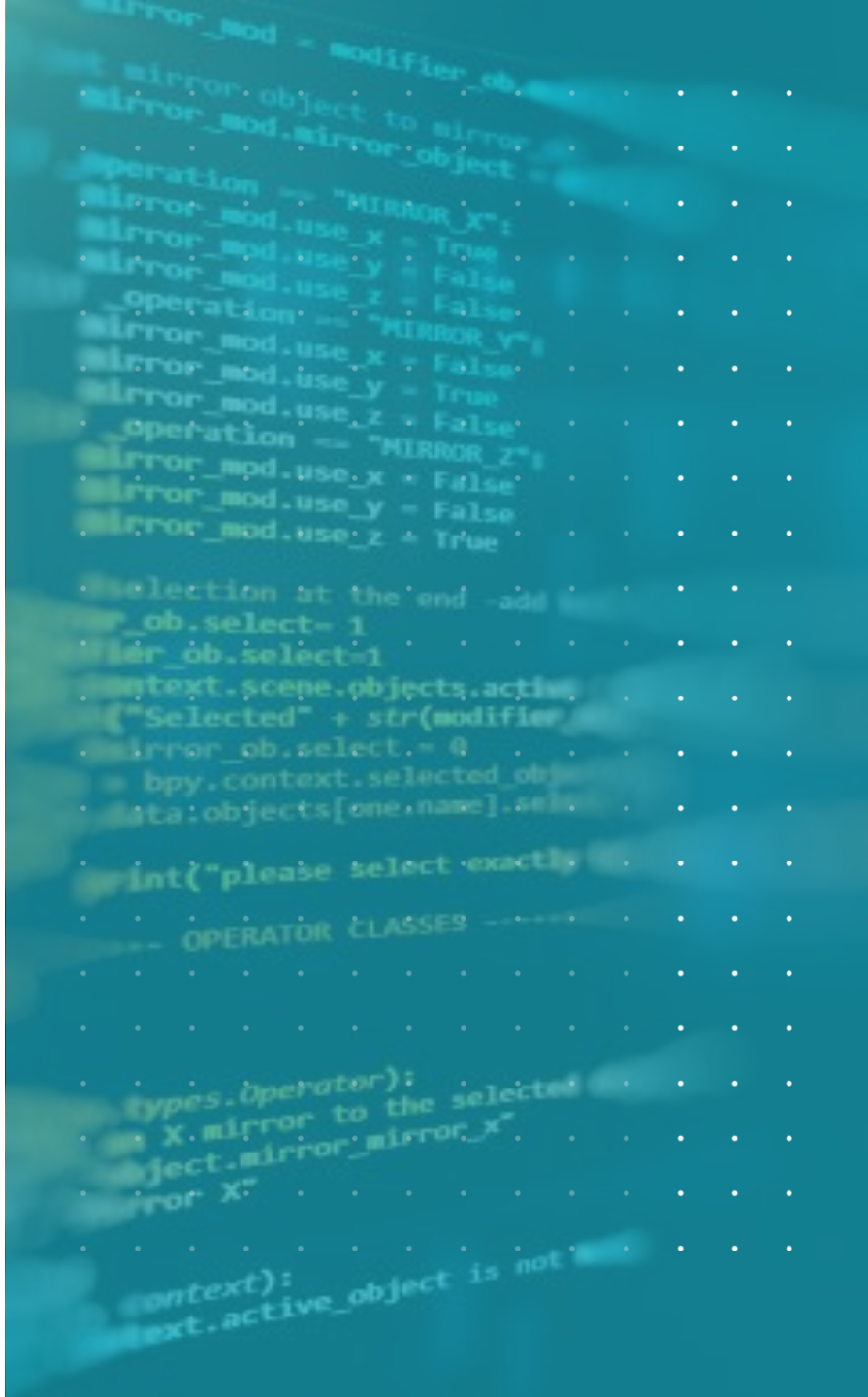
```
> ld -o hello.exe hello.o ...libraries...
```

Verbose Mode -V

```
soroosh@ssoroosh-pc:/mnt/c/Users/ssoroosh/temp$ g++ -v -g -shared -o Hello.exe Hello.o
Using built-in specs.
COLLECT_GCC=g++
COLLECT_LTO_WRAPPER=/usr/lib/gcc/x86_64-linux-gnu/9/lto-wrapper
OFFLOAD_TARGET_NAMES=nvptx-none:hsa
OFFLOAD_TARGET_DEFAULT=1
Target: x86_64-linux-gnu
Configured with: ../src/configure -v --with-pkgversion='Ubuntu 9.3.0-10ubuntu2' --with-bugurl=file:///usr/share/doc/gcc-9/README.Bugs --enable-languages=c,ada,c++,go,brig,d,fortran,objc,obj-c++,gm2 --prefix=/usr --with-gcc-major-version-only --program-suffix=-9 --program-prefix=x86_64-linux-gnu- --enable-shared --enable-linker-build-id --libexecdir=/usr/lib --without-included-gettext --enable-threads=posix --libdir=/usr/lib --enable-nls --enable-clocale=gnu --enable-libstdcxx-debug --enable-libstdcxx-time=yes --with-default-libstdcxx-abi=new --enable-gnu-unique-object --disable-vtable-verify --enable-plugin --enable-default-pie --with-system-zlib --with-target-system-zlib=auto --enable-objc-gc=auto --enable-multiarch --disable-werror --with-arch-32=i686 --with-abi=m64 --with-multilib-list=m32,m64,mx32 --enable-multilib --with-tune=generic --enable-offload-targets=nvptx-none,hsa --without-cuda-driver --enable-checking=release --build=x86_64-linux-gnu --host=x86_64-linux-gnu --target=x86_64-linux-gnu
Thread model: posix
gcc version 9.3.0 (Ubuntu 9.3.0-10ubuntu2)
COMPILER_PATH=/usr/lib/gcc/x86_64-linux-gnu/9/:/usr/lib/gcc/x86_64-linux-gnu/9/:/usr/lib/gcc/x86_64-linux-gnu/:/usr/lib/gcc/x86_64-linux-gnu/9/:/usr/lib/gcc/x86_64-linux-gnu/
LIBRARY_PATH=/usr/lib/gcc/x86_64-linux-gnu/9/:/usr/lib/gcc/x86_64-linux-gnu/9/../../../x86_64-linux-gnu/:/usr/lib/gcc/x86_64-linux-gnu/9/../../../../lib/:/lib/x86_64-linux-gnu/:/lib/../lib/:/usr/lib/x86_64-linux-gnu/:/usr/lib/../lib/:/usr/lib/gcc/x86_64-linux-gnu/9/../../../:/lib/:/usr/lib/
COLLECT_GCC_OPTIONS='-v' '-g' '-shared' '-o' 'Hello.exe' '-shared-libgcc' '-mtune=generic' '-march=x86-64'
 /usr/lib/gcc/x86_64-linux-gnu/9/collect2 -plugin /usr/lib/gcc/x86_64-linux-gnu/9/liblto_plugin.so -plugin-opt=/usr/lib/gcc/x86_64-linux-gnu/9/lto-wrapper -plugin-opt=-fresolution=/tmp/cceQnkzs.res -plugin-opt=-pass-through=-lgcc_s -plugin-opt=-pass-through=-lc -plugin-opt=-pass-through=-lgcc_s --build-id --eh-frame-hdr -m elf_x86_64 --hash-style=gnu --as-needed -shared -z relro -o Hello.exe /usr/lib/gcc/x86_64-linux-gnu/9/../../../x86_64-linux-gnu/crti.o /usr/lib/gcc/x86_64-linux-gnu/9/crtbeginS.o -L/usr/lib/gcc/x86_64-linux-gnu/9 -L/usr/lib/gcc/x86_64-linux-gnu/9/../../../x86_64-linux-gnu -L/usr/lib/gcc/x86_64-linux-gnu/9/../../../../lib -L/lib/x86_64-linux-gnu -L/lib/../lib -L/usr/lib/x86_64-linux-gnu -L/usr/lib/../lib -L/usr/lib/gcc/x86_64-linux-gnu/9/../../.. Hello.o -lstdc++ -lm -lgcc_s -lc -lgcc_s /usr/lib/gcc/x86_64-linux-gnu/9/crtendS.o /usr/lib/gcc/x86_64-linux-gnu/9/../../../x86_64-linux-gnu/crtn.o
COLLECT_GCC_OPTIONS='-v' '-g' '-shared' '-o' 'Hello.exe' '-shared-libgcc' '-mtune=generic' '-march=x86-64'
```

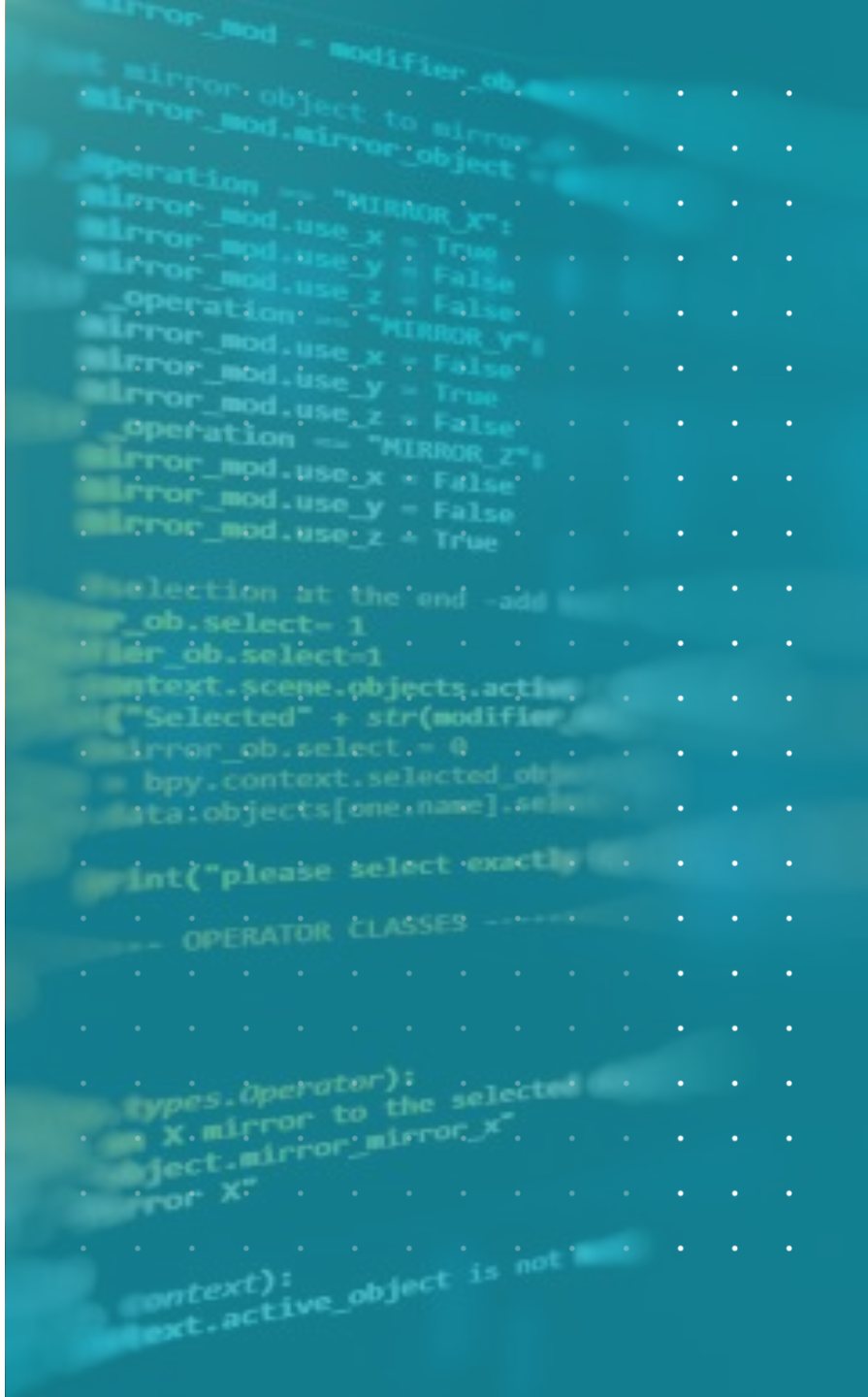
Library

- A library is a collection of pre-compiled object files that can be linked into your programs via the linker.
- When your program is linked against a static library, the machine code of external functions used in your program is copied into the executable.
- When your program is linked against a shared library, only a small table is created in the executable. Before the executable starts running, the operating system loads the machine code needed for the external functions



Dynamic Linking

- A shared library has file extension of ".so" (shared objects) in Unixes or ".dll" (dynamic link library) in Windows.
- Dynamic linking makes executable files smaller and saves disk space
- Often save memory
- The shared library codes can be upgraded without the need to recompile your program.
- Because of the advantage of dynamic linking, GCC, by default, links to the shared library if it is available.
-

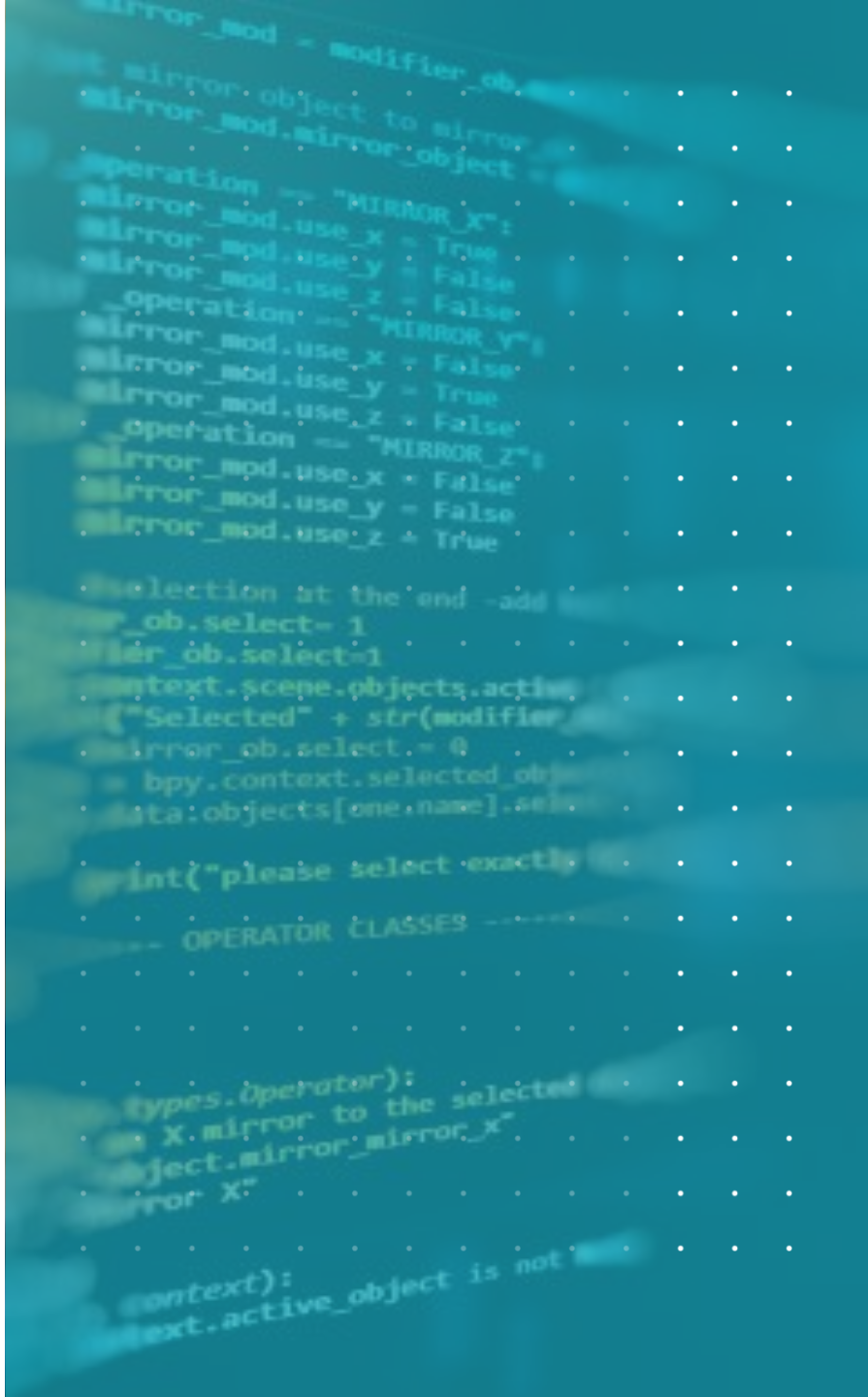


NM list symbols from objects file

```
soroosh@ssoroosh-pc:/mnt/c/Users/ssoroosh/temp$ nm Hello.exe
0000000000003e90 d _DYNAMIC
0000000000004000 d _GLOBAL_OFFSET_TABLE_
                 w _ITM_deregisterTMCloneTable
                 w _ITM_registerTMCloneTable
00000000000020a0 r __FRAME_END__
0000000000002000 r __GNU_EH_FRAME_HDR
0000000000004020 d __TMC_END__
                 w __cxa_finalize
00000000000010b0 t __do_global_dtors_aux
0000000000003e88 d __do_global_dtors_aux_fini_array_entry
0000000000004018 d __dso_handle
0000000000003e80 d __frame_dummy_init_array_entry
                 w __gmon_start__
0000000000001108 t _fini
0000000000001000 t _init
0000000000004020 b completed.8059
0000000000001040 t deregister_tm_clones
00000000000010f0 t frame_dummy
00000000000010f9 T main
0000000000001070 t register_tm_clones
```

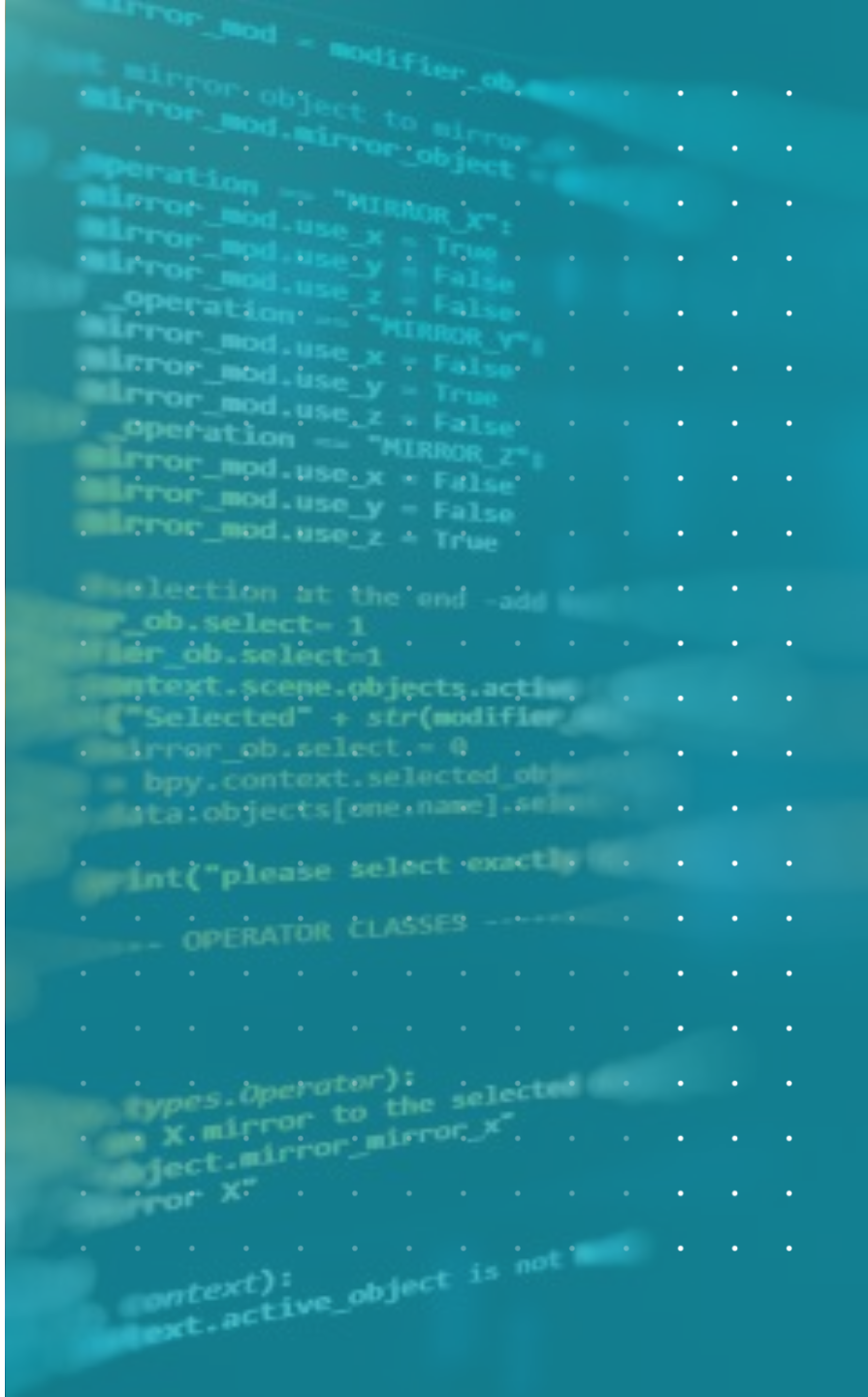
Searching for Header Files and Libraries

- The include-paths are specified via -Idir option –I
- The library-path is specified via -Ldir option –L
- you also have to specify the library name. –l



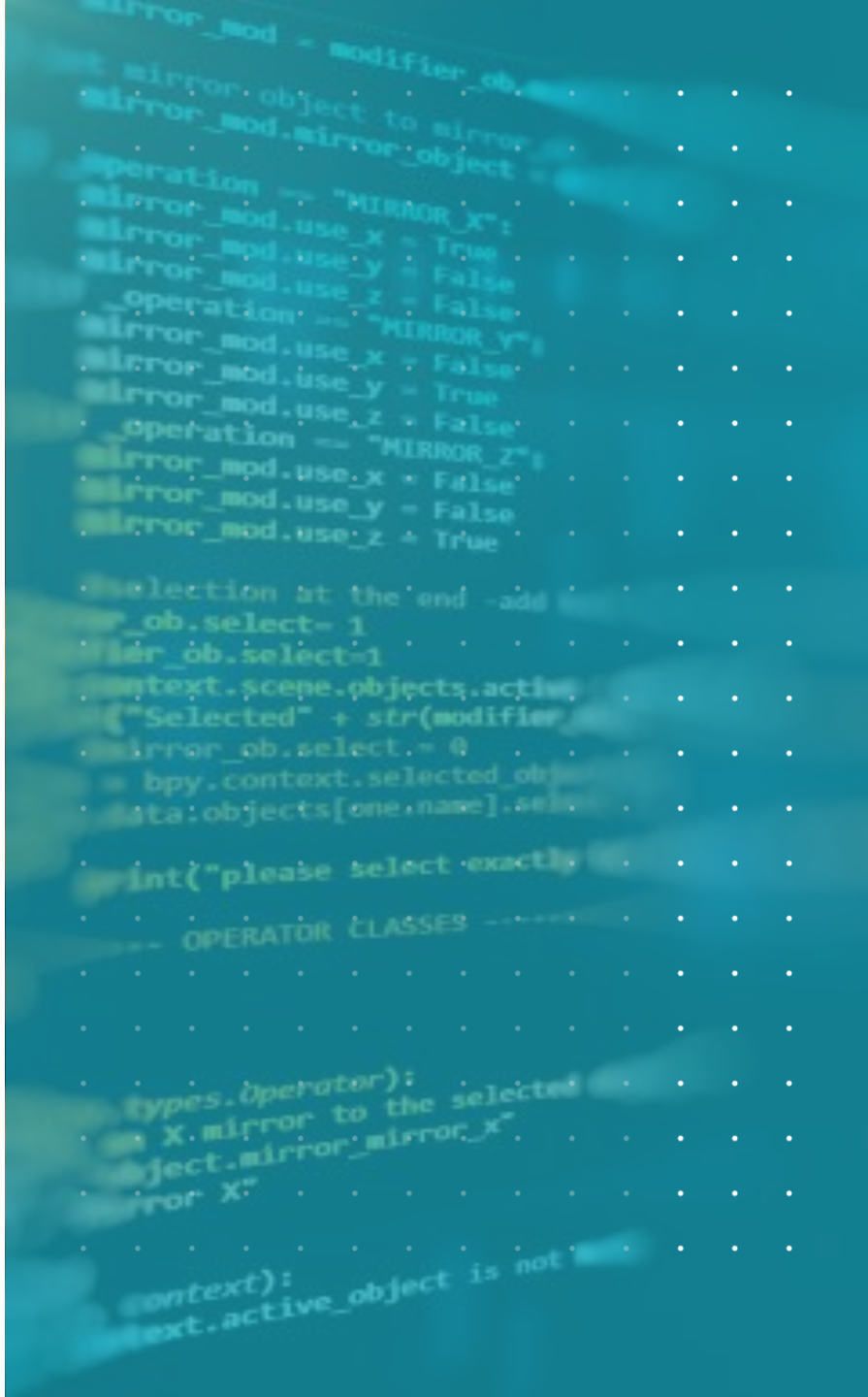
"file" Utility - Determine File Type

```
$ gcc -c hello.c
$ gcc -o hello.exe hello.o

$ file hello.c
hello.c: C source, ASCII text, with CRLF line terminators

$ file hello.o
hello.o: data

> file hello.exe
hello.exe: PE32 executable (console) x86-64, for MS Windows
```



"ldd" Utility - List Dynamic-Link Libraries

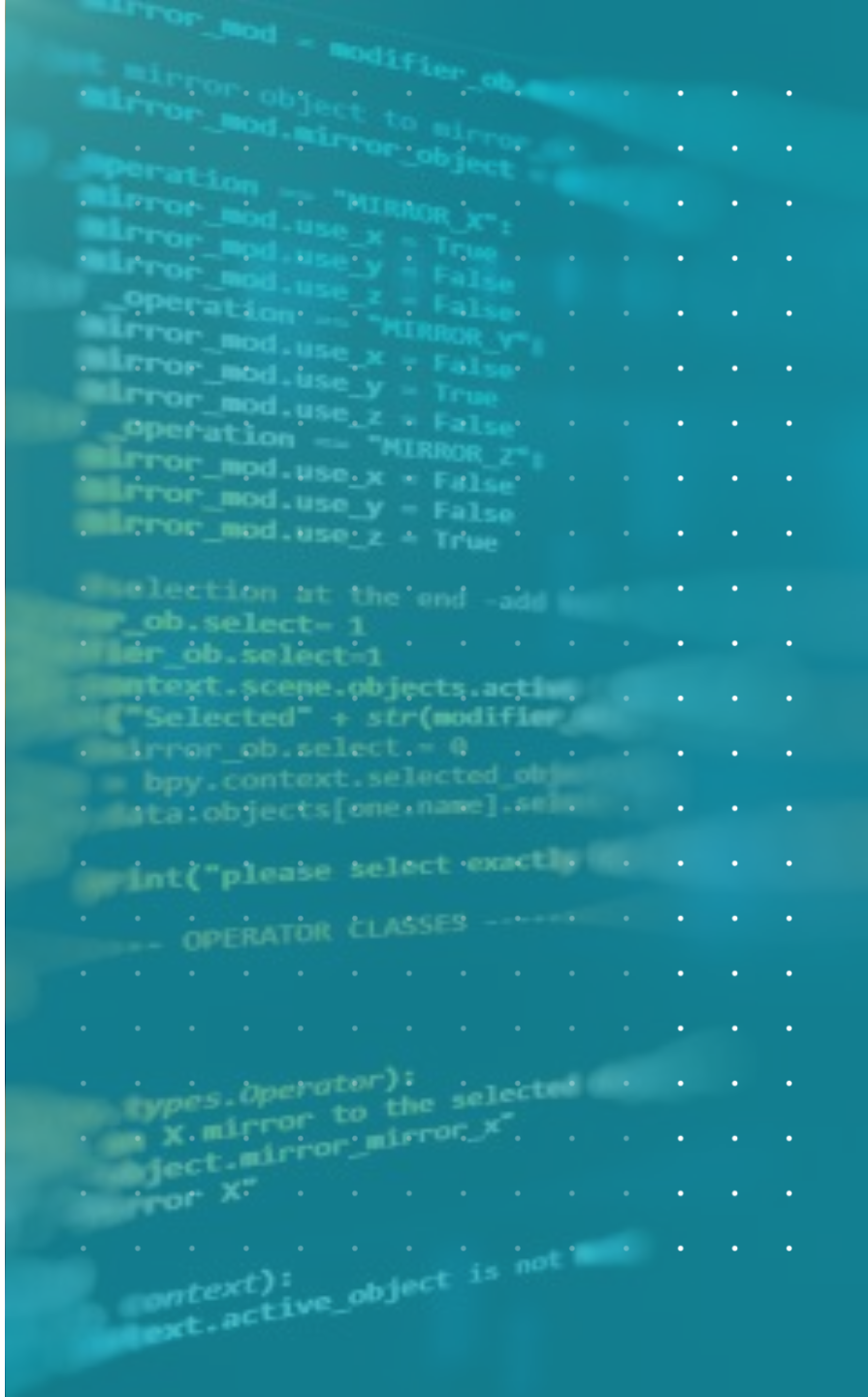
```
soroosh@ssoroosh-pc:/mnt/c/Users/ssoroosh/temp$ ldd ./Hello.exe
        linux-vdso.so.1 (0x00007fffc6f4d000)
        libc.so.6 => /lib/x86_64-linux-gnu/libc.so.6 (0x00007f1659b90000)
        /lib64/ld-linux-x86-64.so.2 (0x00007f1659da7000)
```




GNU Make

First Makefile By Example

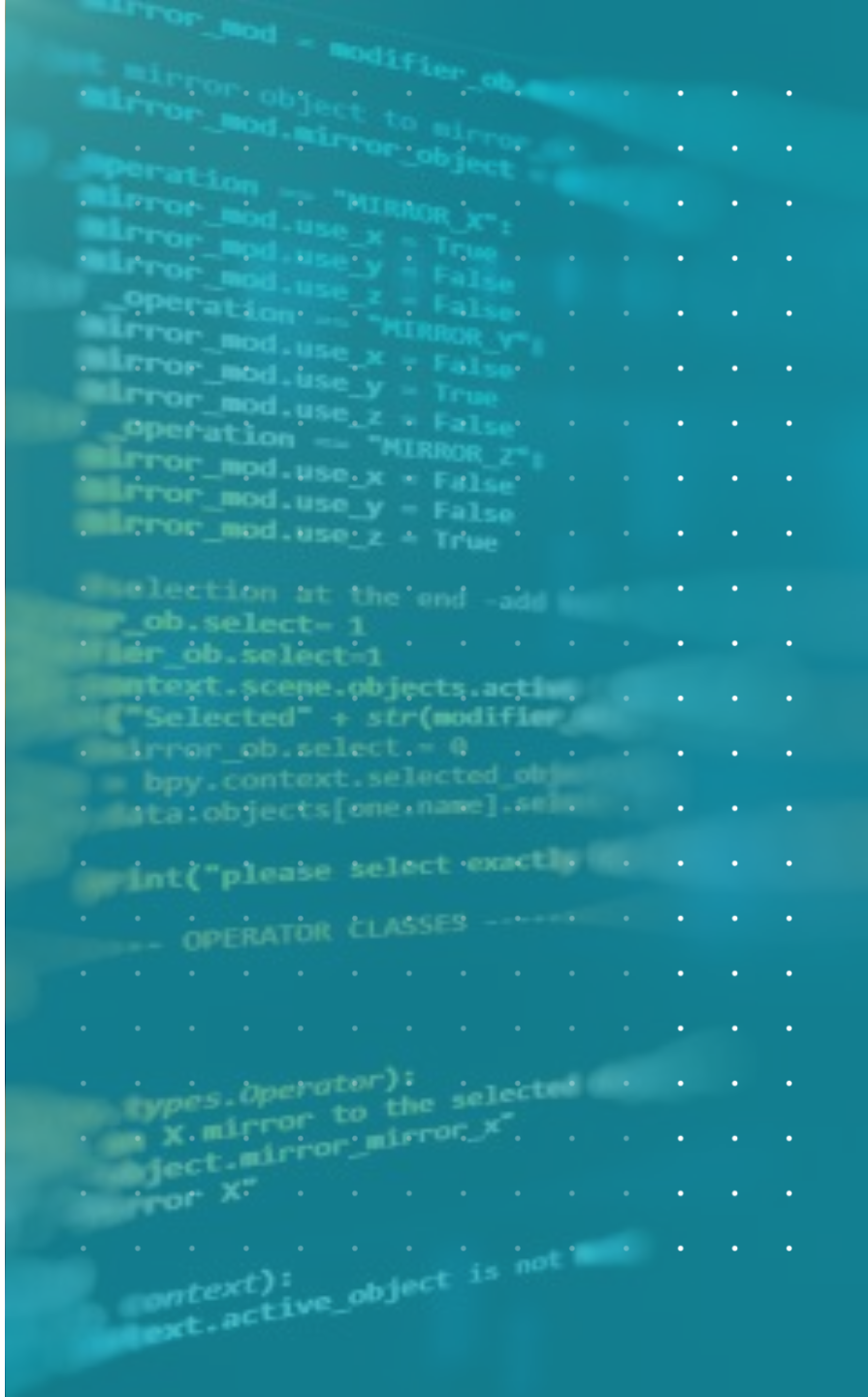
```
all: hello.exe

hello.exe: hello.o
	gcc -o hello.exe hello.o

hello.o: hello.c
	gcc -c hello.c

clean:
	rm hello.o hello.exe
```



Structure

- Running make without argument starts the target "all" in the makefile.
- A makefile consists of a set of rules.
- A rule consists of ۳ parts: a target, a list of pre-requisites and a command
- The *command* must be preceded by a tab (NOT spaces)

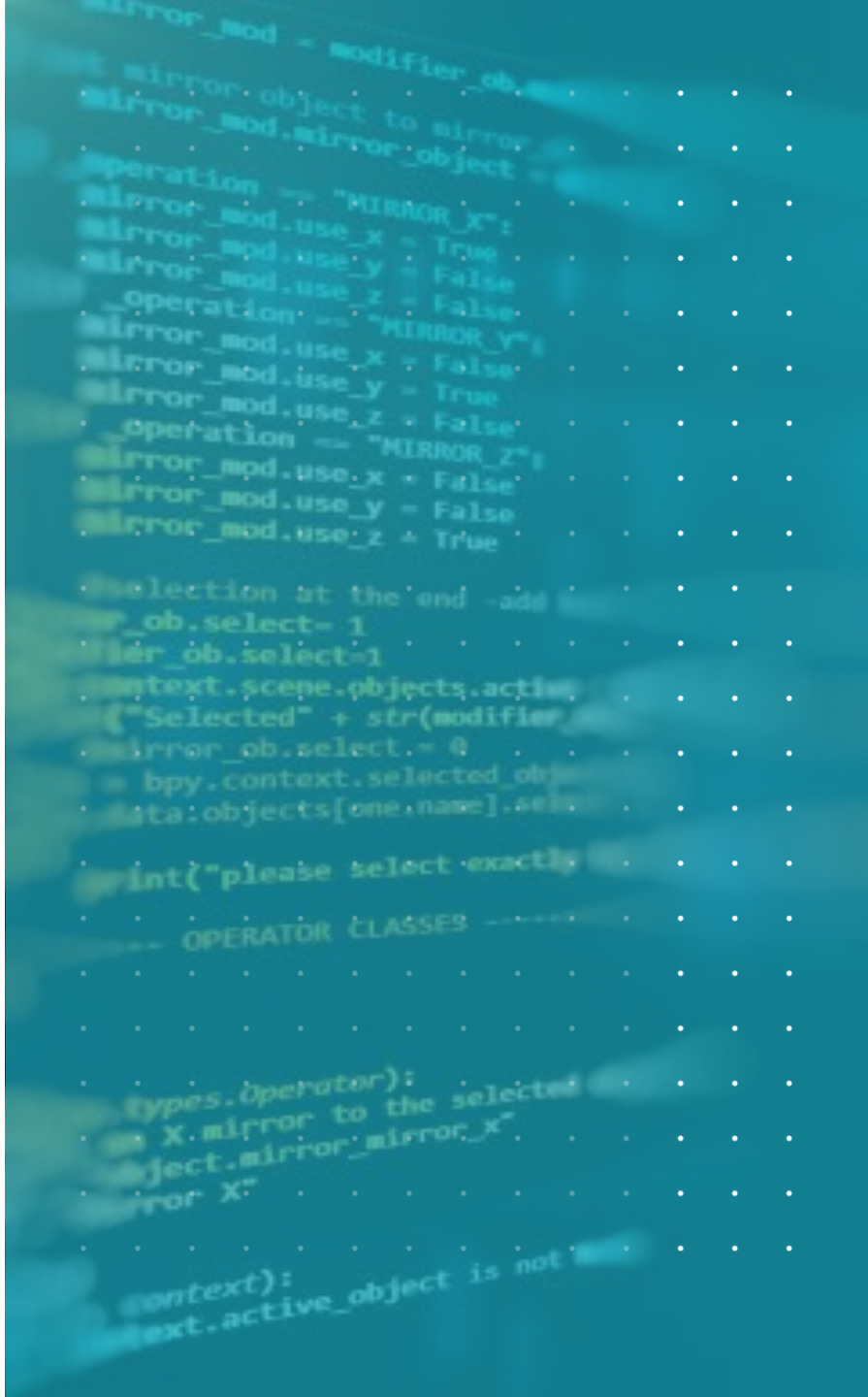
```
target: pre-req-1 pre-req-2 ...
        command
```



Golden Behavior

- When make is asked to evaluate a rule, it begins by finding the files in the prerequisites. If any of the prerequisites has an associated rule, make attempts to update those first.

- if the pre-requisite is not newer than than target, the command will not be run



کمی بیشتر

Phony Targets

A target that does not represent a file is called a phony target.

Phony target is always out-of-date and its command will be run.

The standard phony targets are: all, clean, install

Syntax of Rules

The rules are usually organized in such as way the more general rules come first.

Comment & Continuation

#

\

Variables

Automatic Variables

Automatic variables are set by make after a rule is matched.

- $*: the target filename without the file extension.
- $<: the first prerequisite filename.
- $@: the target filename.
- $^: the filenames of all the prerequisites, separated by spaces, discard duplicates.
- $+: similar to $^, but includes duplicates.
- $?: the names of all prerequisites that are newer than the target, separated by spaces.

$ () or {}

$(CC)

${CC}

Single character variables do not need the parentheses.

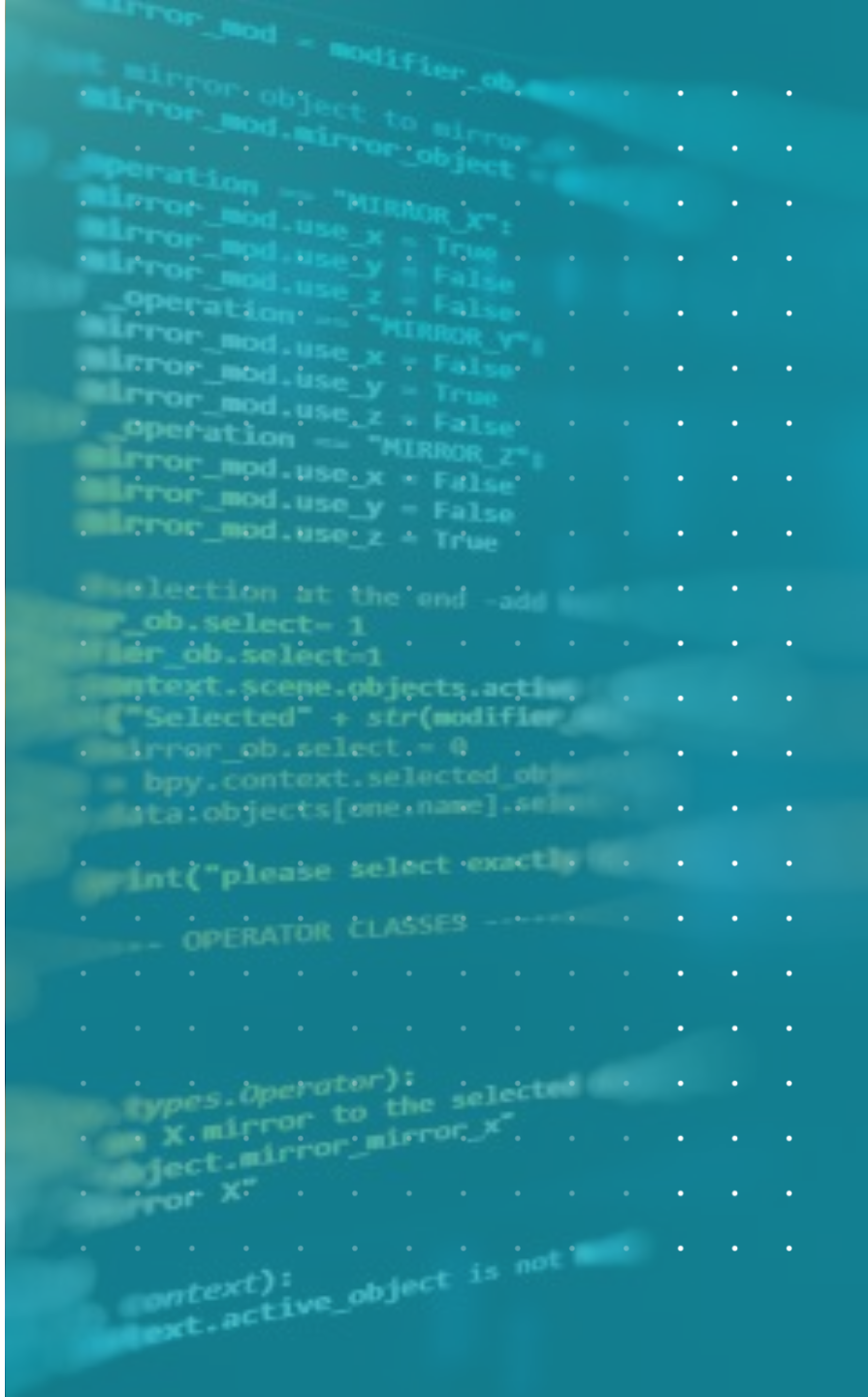
Virtual Path - VPATH & vpath

- You can use VPATH (uppercase) to specify the directory to search for dependencies and target files.

```
# Search for dependencies and targets from "src" and "include" directories
# The directories are separated by space
VPATH = src include
```

- You can also use vpath (lowercase) to be more precise about the file type and its search directory.

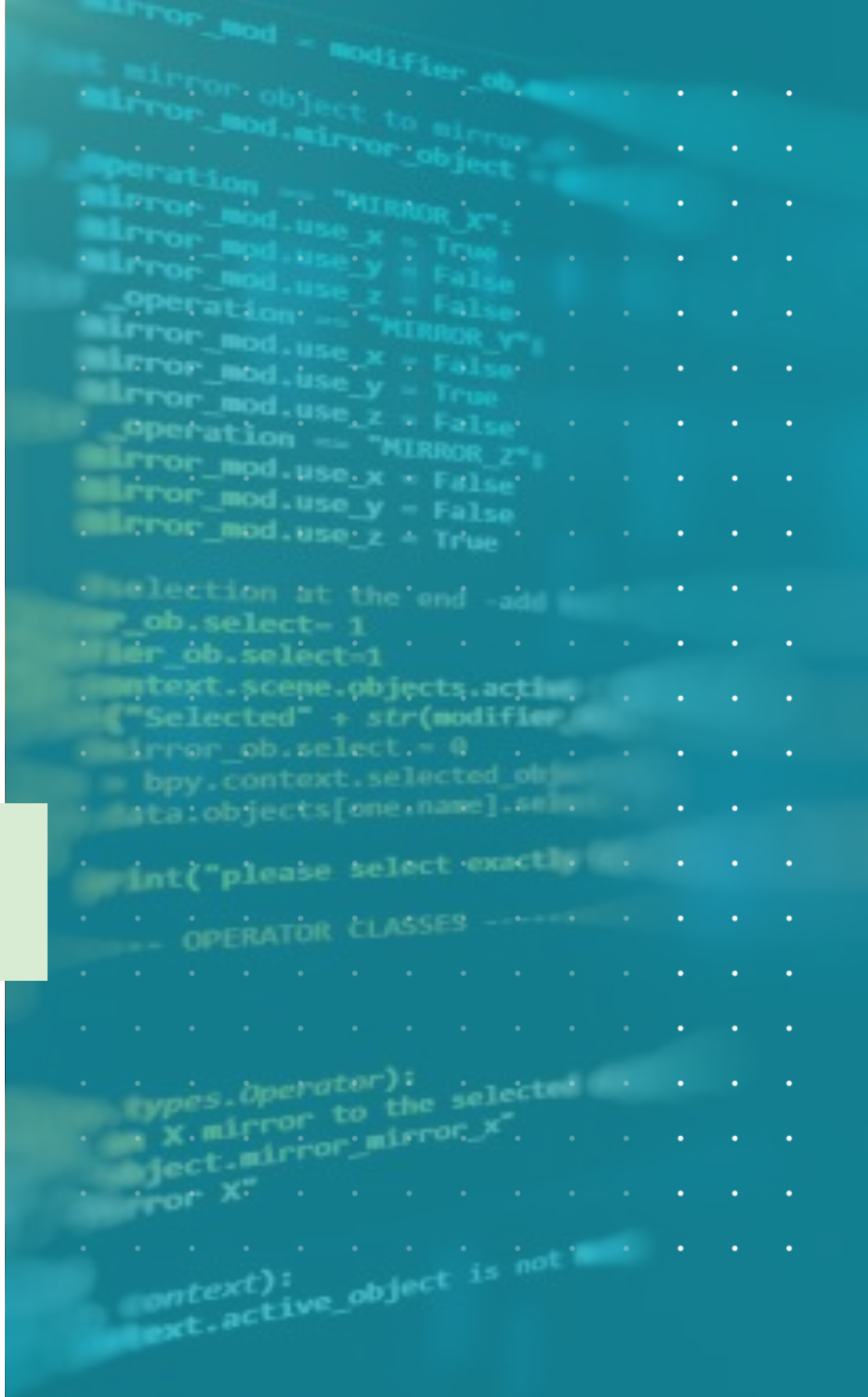
```
# Search for .c files in "src" directory; .h files in "include" directory
# The pattern matching character '%' matches filename without the extension
vpath %.c src
vpath %.h include
```



Pattern Rules

- A pattern rule, which uses pattern matching character '%' as the filename
- can be applied to create a target, if there is no explicit rule.

```
# Applicable for create executable (without extension) from object .o object file
# $^ matches all the pre-requisites (no duplicates)
%: %.o
$(LINK.o) $^ $(LOADLIBES) $(LDLIBS) -o $@
```



```
infodir = $(prefix)/info

#### End of system configuration section. ####

SRCS_C  = tar.c create.c extract.c buffer.c   \
          getoldopt.c update.c gnu.c mangle.c \
          version.c list.c names.c diffarch.c \
          port.c wildmat.c getopt.c getopt1.c \
          regex.c
SRCS_Y  = getdate.y
SRCS    = $(SRCS_C) $(SRCS_Y)
OBJS    = $(SRCS_C:.c=.o) $(SRCS_Y:.y=.o) $(RTAPELIB)
AUX =   README COPYING ChangeLog Makefile.in  \
        makefile.pc configure configure.in \
        tar.texinfo tar.info* texinfo.tex \
        tar.h port.h open3.h getopt.h regex.h \
        rmt.h rmt.c rtapelib.c alloca.c \
        msd_dir.h msd_dir.c tcexparg.c \
        level-0 level-1 backup-specs testpad.c

.PHONY: all
all:    tar rmt tar.info

tar:    $(OBJS)
        $(CC) $(LDFLAGS) -o $@ $(OBJS) $(LIBS)

rmt:    rmt.c
        $(CC) $(CFLAGS) $(LDFLAGS) -o $@ rmt.c

tar.info: tar.texinfo
        makeinfo tar.texinfo

.PHONY: install
install: all
        $(INSTALL) tar $(bindir)/$(binprefix)tar
        -test ! -f rmt || $(INSTALL) rmt /etc/rmt
        $(INSTALLDATA) $(srcdir)/tar.info* $(infodir)
```

از توجه شما
سپاسگزاریم

مهسان
تکیه‌گاه شما
در دنیای هوشمند