Approach to Build Automation

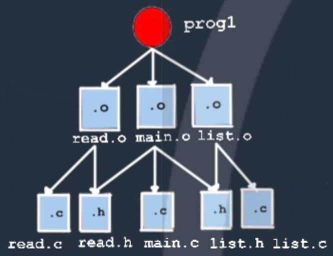
Advantages

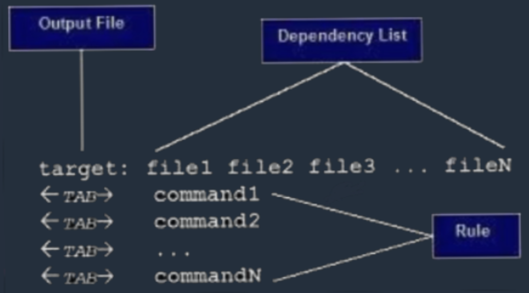
- Makefiles can be a lot handier in large software projects, as using Makefiles can substantially reduce build times if only a few source files have changed, and help automate the process.

- Automation
- Simplicity
- Almost no dependencies

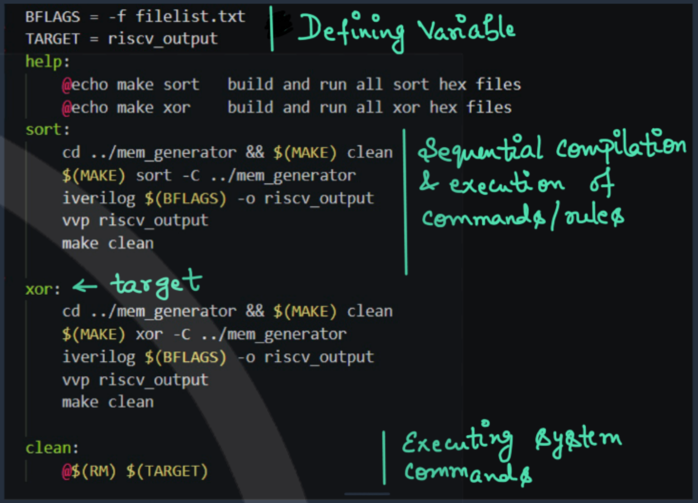
Now, let me quickly walk you through the basic implementation of makefiles & how they work

No! Makefiles use a dependency graph. If none of the files that are prerequisites have been changed since the last time the program was compiled, no actions take place





```
target: file1 file2 file3 ... fileN
←TAB→   command1
←TAB→   command2
←TAB→   ...
←TAB→   commandN
```



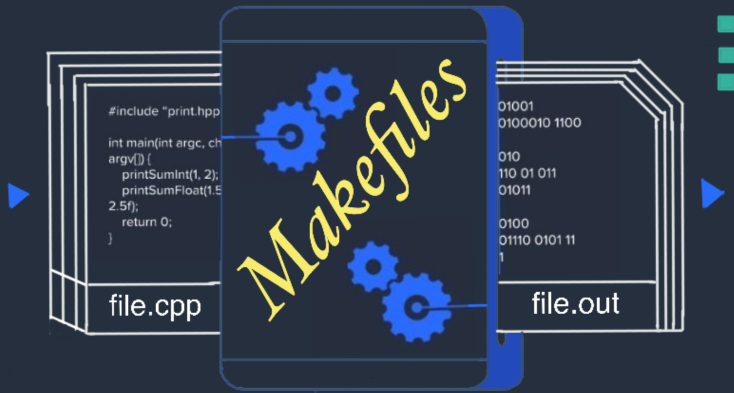
```
BFLAGS = -f filelist.txt
TARGET = riscv_output
help:
	@echo make sort   build and run all sort hex files
	@echo make xor    build and run all xor hex files
sort:
	cd ../mem_generator && $(MAKE) clean
	$(MAKE) sort -C ../mem_generator
	iverilog $(BFLAGS) -o riscv_output
	vvp riscv_output
	make clean

xor:
	cd ../mem_generator && $(MAKE) clean
	$(MAKE) xor -C ../mem_generator
	iverilog $(BFLAGS) -o riscv_output
	vvp riscv_output
	make clean

clean:
	@$(RM) $(TARGET)
```

Defining Variable
Sequential compilation & execution of commands/rules
← target
Executing system commands

- Target: Action to be executed by the rule
- Dependencies: Needed to exist to satisfy the target
- Commands: Sequence of commands to be executed



If only one file is modified, will it compile all the files all over again?

Thank you, sensei! Now I will be able to compile & execute programs more efficiently with this handy automation tool. Enlightenment achieved!

- The makefile is recursively carried out (with dependency prepared before each target depending upon them) until everything has been updated (that requires updating) and the primary/ultimate target is complete

- These instructions with their dependencies are specified in a makefile.

I'm tired of compiling & executing multiple dependent files whenever I change little bit of code.
But I have a lot of dependent files, to be executed in a specific order & spread around different directories. Can makefiles do that?

Haven't you heard of makefiles? I always use them to automate my files sequentially. A makefile is a file containing a set of directives. used by a make build automation tool to generate a target/goal.

Yes! That is the case indeed. In makefile we can define the exact order in which we want our rules/commands to execute.

Aditya Tripathi : 18116010
Kushagra Sharma : 18116091

<u>Material References</u>

- https://speakerdeck.com/pantuza/makefile-automating-daily-tasks-and-simplifying-gettings-started
- http://www.emblogic.com/blog/06/what-does-a-makefile-contain
- https://upload.wikimedia.org/wikipedia/en/thumb/2/22/Heckert_GNU_white.svg/1200px-Heckert_GNU_white.svg.png
- https://www.toptal.com/c-plus-plus/c-plus-plus-understanding-compilation