Make

Antes disso: Exercício

- Se você usa Windows
  - Instale o WSL2 de acordo com o vídeo publicado
  - A partir daí, você pode “fingir” que usa Ubuntu

- Use seu gerenciador de pacotes para instalar as ferramentas necessárias
  - GCC, make, editor de texto (vscode), valgrind
  - Comunique erros caso aconteçam!

Build

- Processo de transformar uma base de código em software
- Usa um grande conjunto de ferramentas
- Pode ficar bem complexo
- Costumava levar dias para alguns projetos

Sistema de Build (Make)

- Automatiza os comandos necessários
- Centraliza informações importantes
- Pode ser redistribuído para outros usuários
- Exemplos: Make, Gradle, Scons, Cargo

Make

- Mais um do Bell Labs em 1976
- Pouco estruturado
- Sistema de *targets* e *dependencies*
- Usa apenas o arquivo Makefile

Regras

- Componente básico do Makefile

```
main.o: main.c
   gcc -c main.c -o main.o
```

- Três componentes:
  - *target*: main.o
  - Dependências: main.c
  - Comando: a compilação

Sobre Regras

- Você é livre para definir o que cada regra faz
- Regras não executam se “não precisarem”
- Esperam gerar um arquivo com nome da regra (o target)
- Algumas são convenções: all, clean, run

Variáveis

- Pra não ter que escrever coisas várias vezes
- Sintaxe semelhante ao Bash

```
CFLAGS = -std=c99 -Wall -Wextra -O2
gcc -c $(CFLAGS) main.c -o main.o
```

- Sem aspas, espaço no “=”
- Apenas textuais

Variáveis 2

- Podem ser usadas pra reduzir esforço
  - Mudando compilador, flags, em todas as regras
- Centralizam decisões do Makefile (reduzindo erros)
- Varíaveis automáticas tem valores especiais
  - $^: dependencias
  - $@: target

Pattern Matching

- Podem ser usados pra criar regras genéricas
  - Evita ter que criar uma regra pra cada arquivo

```
%.o: %.c
```

- Executa para todo arquivo terminando em .c, gerando um .o
- Pode usar as variáveis automáticas

Alguns detalhes

- Regra .PHONY
  - Define que nome da regra não remete a um arquivo
  - Útil para: all, clean, run, teste

- Existem comandos em make
  - patsubst faz pattern matching pra inserir prefixos/sufixos
  - sort ordena uma lista

Demonstração

Vai dar certo, confia!

Depois disso: Exercício

- Se você usa Windows
  - Instale o WSL2 de acordo com o vídeo publicado
  - A partir daí, você pode “fingir” que usa Ubuntu

- Use seu gerenciador de pacotes para instalar as ferramentas necessárias
  - GCC, make, editor de texto (vscode), valgrind
  - Comunique erros caso aconteçam!