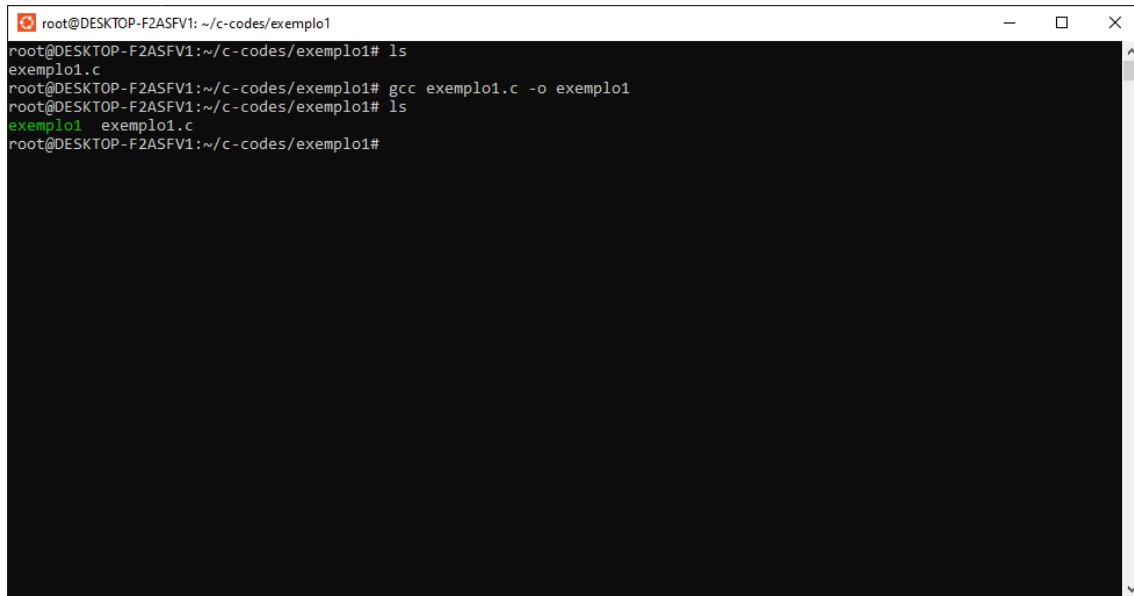


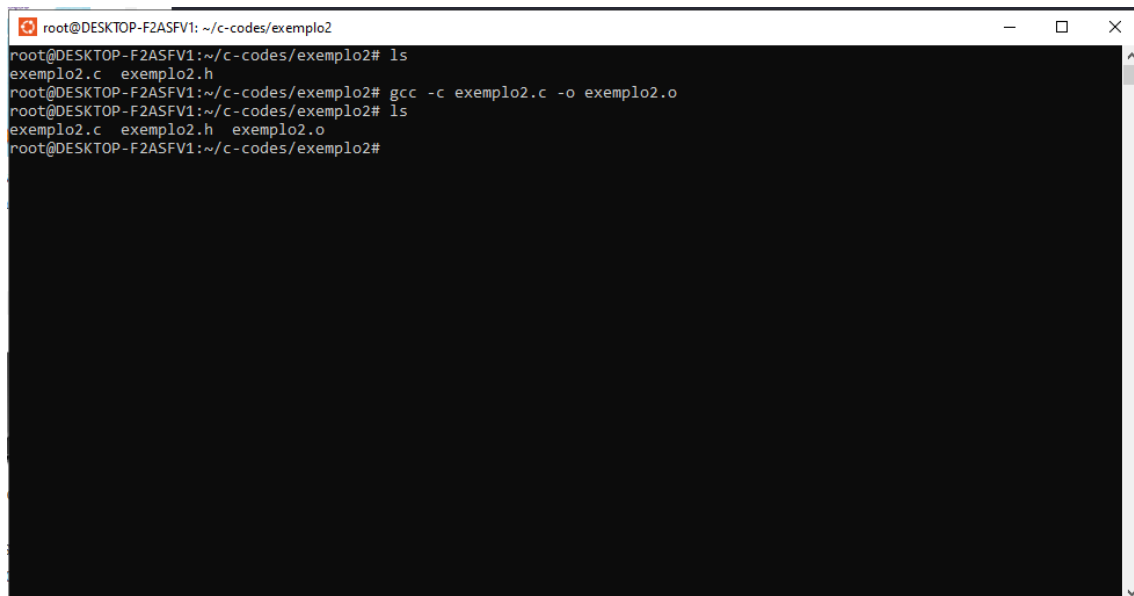
2)Vídeo 1:

Exemplo 1: Compilar um arquivo “.c” utilizando o comando: gcc nome_do_arquivo.c -o nome_de_saída. Executar o arquivo com: ./nome_de_saída.



```
root@DESKTOP-F2ASFV1: ~/c-codes/exemplo1
root@DESKTOP-F2ASFV1:~/c-codes/exemplo1# ls
exemplo1.c
root@DESKTOP-F2ASFV1:~/c-codes/exemplo1# gcc exemplo1.c -o exemplo1
root@DESKTOP-F2ASFV1:~/c-codes/exemplo1# ls
exemplo1  exemplo1.c
root@DESKTOP-F2ASFV1:~/c-codes/exemplo1#
```

Exemplo 2: Compilar um arquivo .c para te rum arquivo objeto de saída utilizando o comando: gcc -c nome_do_arquivo.c -o nome_de_saída.o.



```
root@DESKTOP-F2ASFV1: ~/c-codes/exemplo2
root@DESKTOP-F2ASFV1:~/c-codes/exemplo2# ls
exemplo2.c  exemplo2.h
root@DESKTOP-F2ASFV1:~/c-codes/exemplo2# gcc -c exemplo2.c -o exemplo2.o
root@DESKTOP-F2ASFV1:~/c-codes/exemplo2# ls
exemplo2.c  exemplo2.h  exemplo2.o
root@DESKTOP-F2ASFV1:~/c-codes/exemplo2#
```

Exemplo 3: Gerar um arquivo objeto a partir de um arquivo .c e salvar esse arquivo em um diretório utilizando o comando: gcc -c nome_do_arquivo.c -o obj/nome_de_saída.o.

```
root@DESKTOP-F2ASFV1: ~/c-codes/emplo2
root@DESKTOP-F2ASFV1:~/c-codes/emplo2# mkdir obj
root@DESKTOP-F2ASFV1:~/c-codes/emplo2# ls
emplo2.c  emplo2.h  obj
root@DESKTOP-F2ASFV1:~/c-codes/emplo2# gcc -c emplo2.c -o obj/emplo2.o
root@DESKTOP-F2ASFV1:~/c-codes/emplo2# tree .
.
├── emplo2.c
├── emplo2.h
└── obj
    └── emplo2.o

1 directory, 3 files
root@DESKTOP-F2ASFV1:~/c-codes/emplo2#
```

Exemplo 4: Gerar um arquivo objeto com includes em diretórios diferentes utilizando o seguinte comando: `gcc -c nome_do_arquivo.c -I diretorio_de_includes/ -o diretorio_de_saida/nome_de_saida.o`.

```
root@DESKTOP-F2ASFV1: ~/c-codes/emplo2
root@DESKTOP-F2ASFV1:~/c-codes/emplo2# tree .
.
├── emplo2.c
├── includes
│   └── emplo2.h
└── obj

2 directories, 2 files
root@DESKTOP-F2ASFV1:~/c-codes/emplo2# gcc -c emplo2.c -I includes/ -o obj/emplo2.o
root@DESKTOP-F2ASFV1:~/c-codes/emplo2# tree .
.
├── emplo2.c
├── includes
│   └── emplo2.h
└── obj
    └── emplo2.o

2 directories, 3 files
root@DESKTOP-F2ASFV1:~/c-codes/emplo2#
```

Vídeo 2:

Exemplo 1: Gerar source lendo arquivos cada um em um diretório

```
root@DESKTOP-F2ASFV1: ~/Video_02/exemplo1
root@DESKTOP-F2ASFV1:~/Video_02/exemplo1# tree .
.
├── includes
│   └── exemplo2.h
├── obj
├── src
│   └── exemplo2.c
└── 3 directories, 2 files

root@DESKTOP-F2ASFV1:~/Video_02/exemplo1# gcc -c src/exemplo2.c -I ./includes/ -o obj/exemplo2.o
root@DESKTOP-F2ASFV1:~/Video_02/exemplo1# tree .
.
├── includes
│   └── exemplo2.h
├── obj
│   └── exemplo2.o
├── src
│   └── exemplo2.c
└── 3 directories, 3 files

root@DESKTOP-F2ASFV1:~/Video_02/exemplo1#
```

Exemplo 2: Gerar um arquivo executável no diretório bin tendo já um arquivo objeto de include.

```
root@DESKTOP-F2ASFV1: ~/Video_02/exemplo2
root@DESKTOP-F2ASFV1:~/Video_02/exemplo2# gcc ./app/app.c ./obj/exemplo2.o -I ./inc/ -o ./bin/app
/usr/bin/ld: cannot find ./obj/exemplo2.o: No such file or directory
collect2: error: ld returned 1 exit status
root@DESKTOP-F2ASFV1:~/Video_02/exemplo2# gcc -c src/exemplo2.c -I ./inc/ -o obj/exemplo2.o
root@DESKTOP-F2ASFV1:~/Video_02/exemplo2# gcc ./app/app.c ./obj/exemplo2.o -I ./inc/ -o ./bin/app
root@DESKTOP-F2ASFV1:~/Video_02/exemplo2# tree ./
./
├── app
│   └── app.c
├── bin
│   └── app
├── inc
│   └── exemplo2.h
├── obj
│   └── exemplo2.o
├── src
│   └── exemplo2.c
└── 5 directories, 5 files

root@DESKTOP-F2ASFV1:~/Video_02/exemplo2# ./bin/app
0 valor é:2
root@DESKTOP-F2ASFV1:~/Video_02/exemplo2#
```

Vídeo 3:

Exemplo 1: Utilizar o makefile para poder executar comandos diretamente na bash de forma atualizada

```
root@DESKTOP-F2ASFV1: ~/Video_03/exemplo1
root@DESKTOP-F2ASFV1:~/Video_03/exemplo1# ls
Makefile  app  bin  inc  obj  src
root@DESKTOP-F2ASFV1:~/Video_03/exemplo1# cat Makefile
all:
    tree .
root@DESKTOP-F2ASFV1:~/Video_03/exemplo1# make
tree .
.
├── Makefile
├── app
│   └── app.c
├── bin
├── inc
│   └── exemplo2.h
├── obj
├── src
│   └── exemplo2.c
5 directories, 4 files
root@DESKTOP-F2ASFV1:~/Video_03/exemplo1#
```

Vídeo 4:

Exemplo 1: Compilar os arquivos objetos e o executável utilizando comando no Makefile.

```
root@DESKTOP-F2ASFV1: ~/Video_04/exemplo1
root@DESKTOP-F2ASFV1:~/Video_04/exemplo1# tree .
.
├── Makefile
├── app
│   └── app.c
├── bin
├── inc
│   └── exemplo2.h
├── obj
├── src
│   └── exemplo2.c
5 directories, 4 files
root@DESKTOP-F2ASFV1:~/Video_04/exemplo1# cat Makefile
all:
    gcc -c ./src/exemplo2.c -I ./inc -o ./obj/exemplo2.o
    gcc ./app/app.c ./obj/*.o -I ./inc/ -o ./bin/app

clean:
    rm ./obj/* ./bin/*
root@DESKTOP-F2ASFV1:~/Video_04/exemplo1# make
gcc -c ./src/exemplo2.c -I ./inc -o ./obj/exemplo2.o
gcc ./app/app.c ./obj/*.o -I ./inc/ -o ./bin/app
root@DESKTOP-F2ASFV1:~/Video_04/exemplo1# tree .
.
├── Makefile
├── app
│   └── app.c
├── bin
│   └── app
├── inc
│   └── exemplo2.h
├── obj
│   └── exemplo2.o
├── src
│   └── exemplo2.c
5 directories, 6 files
```

Exemplo 2: Compilar os arquivos utilizando Makefile mas com algumas automações.

```
Selecionar root@DESKTOP-F2ASFV1: ~/Video_04/emplo2
root@DESKTOP-F2ASFV1:~/Video_04/emplo2# tree .
.
├── Makefile
├── app
│   └── app.c
├── bin
├── inc
│   └── exemplo2.h
├── obj
├── src
│   └── exemplo2.c
└──

5 directories, 4 files
root@DESKTOP-F2ASFV1:~/Video_04/emplo2# cat Makefile
SRC = ./src
INC = ./inc
OBJ = ./obj
BIN = ./bin
APP = ./app

all:
    gcc -c $(SRC)/exemplo2.c -I $(INC) -o $(OBJ)/exemplo2.o
    gcc $(APP)/app.c $(OBJ)/*.o -I $(INC) -o $(BIN)/app

run:
    $(BIN)/app

clean:
    rm $(OBJ)/* $(BIN)/*root@DESKTOP-F2ASFV1:~/Video_04/emplo2# make
gcc -c ./src/exemplo2.c -I ./inc -o ./obj/exemplo2.o
gcc ./app/app.c ./obj/*.o -I ./inc -o ./bin/app
root@DESKTOP-F2ASFV1:~/Video_04/emplo2# make run
./bin/app
0 valor é:2
root@DESKTOP-F2ASFV1:~/Video_04/emplo2# tree .
.
├── Makefile
├── app
│   └── app.c
├── bin
│   └── app
├── inc
│   └── exemplo2.h
├── obj
│   └── exemplo2.o
├── src
│   └── exemplo2.c
└──

5 directories, 6 files
root@DESKTOP-F2ASFV1:~/Video_04/emplo2#
```

Exemplo 3: Fazer o all: ter duas dependências, uma para criar os objetos e a outra os executáveis.

```
root@DESKTOP-F2ASFV1: ~/Video_04/exemplo3
root@DESKTOP-F2ASFV1:~/Video_04/exemplo3# cat Makefile
SRC = ./src
INC = ./inc
OBJ = ./obj
BIN = ./bin
APP = ./app

all: libraries applications

libraries:
    gcc -c $(SRC)/exemplo2.c -I $(INC) -o $(OBJ)/exemplo2.o

applications:
    gcc $(APP)/app.c $(OBJ)/*.o -I $(INC) -o $(BIN)/app

run:
    $(BIN)/app

clean:
    rm $(OBJ)/* $(BIN)/*root@DESKTOP-F2ASFV1:~/Video_04/exemplo3#
root@DESKTOP-F2ASFV1:~/Video_04/exemplo3# make
gcc -c ./src/exemplo2.c -I ./inc -o ./obj/exemplo2.o
gcc ./app/app.c ./obj/*.o -I ./inc -o ./bin/app
root@DESKTOP-F2ASFV1:~/Video_04/exemplo3# make run
./bin/app
0 valor é:2
root@DESKTOP-F2ASFV1:~/Video_04/exemplo3# tree .
.
├── Makefile
├── app
│   └── app.c
├── bin
│   └── app
├── inc
│   └── exemplo2.h
├── obj
│   └── exemplo2.o
└── src
    └── exemplo2.c

5 directories, 6 files
root@DESKTOP-F2ASFV1:~/Video_04/exemplo3#
```

Vídeo 5:

Exemplo 1: Passa a diretiva %.o para realizar o comando de compilação de objeto para todos .o

```
root@DESKTOP-F2ASFV1: ~/Video_05/exemplo1
root@DESKTOP-F2ASFV1:~/Video_05/exemplo1# tree .
.
├── Makefile
├── app
│   └── app.c
├── bin
├── inc
│   └── exemplo2.h
├── obj
├── src
│   └── exemplo2.c
└── 5 directories, 4 files

root@DESKTOP-F2ASFV1:~/Video_05/exemplo1# cat Makefile
SRC = ./src
INC = ./inc
OBJ = ./obj
BIN = ./bin
APP = ./app

all: libraries applications

libraries:
    gcc -c $(SRC)/exemplo2.c -I $(INC) -o $(OBJ)/exemplo2.o

applications:
    gcc $(APP)/app.c $(OBJ)/*.o -I $(INC)/ -o $(BIN)/app

%.o: $(SRC)/%.c $(INC)/%.h
    gcc -c $< -I $(INC) -o $(OBJ)/$@

run:
    $(BIN)/app

clean:
    rm $(OBJ)/* $(BIN)/*
root@DESKTOP-F2ASFV1:~/Video_05/exemplo1# make exemplo2.o
gcc -c src/exemplo2.c -I ./inc -o ./obj/exemplo2.o
root@DESKTOP-F2ASFV1:~/Video_05/exemplo1# tree .
.
├── Makefile
├── app
│   └── app.c
├── bin
├── inc
│   └── exemplo2.h
├── obj
│   └── exemplo2.o
├── src
│   └── exemplo2.c
└── 5 directories, 5 files

root@DESKTOP-F2ASFV1:~/Video_05/exemplo1#
```

Exemplo 2: Troca no libraries para chamar os arquivos .o e os compilar.

```
root@DESKTOP-F2ASFV1: ~/Video_05/exemplo2
root@DESKTOP-F2ASFV1:~/Video_05/exemplo2# cat Makefile
SRC = ./src
INC = ./inc
OBJ = ./obj
BIN = ./bin
APP = ./app

all: libraries applications

libraries: exemplo2.o

applications:
    gcc $(APP)/app.c $(OBJ)/*.o -I $(INC)/ -o $(BIN)/app

%.o: $(SRC)/%.c $(INC)/%.h
    gcc -c $< -I $(INC) -o $(OBJ)/$@

run:
    $(BIN)/app

clean:
    rm -rf $(OBJ)/* $(BIN)/*root@DESKTOP-F2ASFV1:~/Video_05/exemplo2# tree .
.
├── Makefile
├── app
│   └── app.c
├── bin
├── inc
│   └── exemplo2.h
├── obj
├── src
│   └── exemplo2.c
└──

5 directories, 4 files
root@DESKTOP-F2ASFV1:~/Video_05/exemplo2# make
gcc -c src/exemplo2.c -I ./inc -o ./obj/exemplo2.o
gcc ./app/app.c ./obj/*.o -I ./inc/ -o ./bin/app
root@DESKTOP-F2ASFV1:~/Video_05/exemplo2# tree .
.
├── Makefile
├── app
│   └── app.c
├── bin
│   └── app
├── inc
│   └── exemplo2.h
├── obj
│   └── exemplo2.o
├── src
│   └── exemplo2.c
└──

5 directories, 6 files
root@DESKTOP-F2ASFV1:~/Video_05/exemplo2#
```