

Commands cheatsheet

Matteo Lugli

June 3, 2023

1 Handling PassManager

Con il comando **opt** si può alterare il comportamento del pass manager di default, specificando manualmente quali sono i passi da svolgere. Si possono caricare dei passi custom in formato .so. Ecco delle direttive da usare all'interno del makefile per generare il file ".so".

```
OPTIMIZER := libTestPass.so
OBJS := $(subst .cpp,.o,$(wildcard lib/*.cpp))

LLVM_VERSION ?= 14

CXXFLAGS := $(shell llvm-config-$(LLVM_VERSION) --cxxflags) -fPIC

all: $(OPTIMIZER)

$(OPTIMIZER): $(OBJS)
$(CXX) -dylib -shared $^ -o $@
```

1.1 Tutorial 01

```
.PHONY: clean
clean:
$(RM) $(OPTIMIZER) $(OBJS)
load:
opt -load-pass-plugin=./libTestPass.so -passes=test-pass $(file) -disable-output
```

In questo caso per eseguire il passo è necessario dare in pasto al make anche il file su cui eseguire l'ottimizzazione:

```
make clean all load file=test/Loop.bc
```

1.2 Tutorial 02

```
.PHONY: clean setup
```

```
clean:
```

```
$(RM) $(OPTIMIZER) $(OBJS)
```

```
setup:
```

```
opt -load-pass-plugin=./libLocalOpts.so -passes=algebraic $(firstrepr) -o $(bcfile)
```

```
llvm-dis $(bcfile) -o $(optrepr)
```

```
.PHONY: auto
```

```
auto:
```

```
make clean all setup firstrepr=./testTransform/Foo.ll bcfile=./testTransform/FooOpt.bc
```

2 OMP

Comando per compilare un file c, linkando la libreria **openmp** e produrre un file intermedio (dopo la procedura di lowering).

```
gcc -fopenmp -fdump-tree-omplower example.c
```