

```

OBSJS = parser.cmo scanner.cmo printer.cmo interpret.cmo pcgsl.cmo
LIBS = unix.cma

TESTS = \
arith1 \
arith2 \
fib \
for1 \
func1 \
gcd \
global1 \
hello \
if1 \
if2 \
if3 \
if4 \
ops1 \
var1 \
while1

TARFILES = Makefile testall.sh scanner.mll parser.mly \
            ast.mli interpret.ml printer.ml pcgsl.ml \
            $(TESTS:%=tests/test-%.mc) \
            $(TESTS:%=tests/test-%.out)

pcgsl : $(OBSJS)
        ocamlc -o pcgsl $(LIBS) $(OBSJS)

.PHONY : test
test : pcgsl testall.sh
        ./testall.sh

scanner.ml : scanner.mll
        ocamllex scanner.mll

parser.ml parser.mli : parser.mly
        ocamlyacc -v parser.mly

%.cmo : %.ml
        ocamlc -c $<

%.cmi : %.mli
        ocamlc -c $<

pcgsl.tar.gz : $(TARFILES)
        tar czf ./pcgsl.tar.gz $(TARFILES:%=./%)

.PHONY : clean
clean :
        rm -f pcgsl parser.ml parser.mli scanner.ml testall.log *.cmo *.cmi *.out pars
er.output

# Generated by ocamldep *.ml *.mli
interpret.cmo: ast.cmi
interpret.cmx: ast.cmi
pcgsl.cmo: scanner.cmo parser.cmi interpret.cmo
pcgsl.cmx: scanner.cmx parser.cmx interpret.cmx
parser.cmo: ast.cmi parser.cmi
parser.cmx: ast.cmi parser.cmi
printer.cmo: ast.cmi
printer.cmx: ast.cmi
scanner.cmo: parser.cmi
scanner.cmx: parser.cmx
parser.cmi: ast.cmi

```