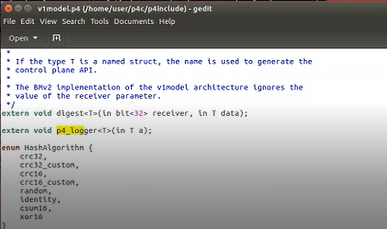
**0615在p4程式裡多一個函式：p4\_logger**

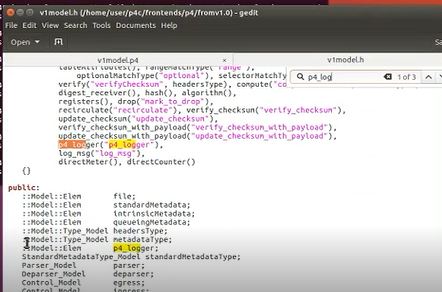
*在開始程式的時候，想把那些東西印出來就可以印出來*

安裝步驟：

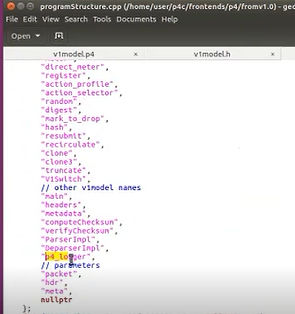
1. 先到<https://github.com/cslev/p4extern>網站
2. 打開終端機，切到p4-test，執行gedit &
3. Open -> other documents -> user -> p4c -> p4include -> v1model.p4
4. 把extern void p4\_logger<T>(in T a);加上



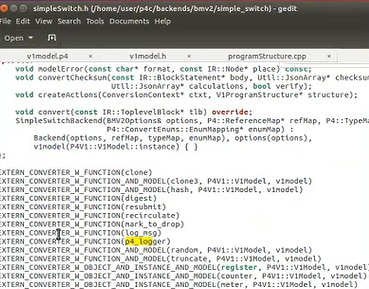
1. 加上以後，繼續加下一個。一樣Open -> other documents
2. user -> p4c -> frontends -> p4 -> fromv1.0 -> v1model.h
3. 加入p4\_logger("p4\_logger"), / ::Model::Elem p4\_logger;



1. 繼續加下一個。Open -> other documents
2. user -> p4c -> frontends -> p4 -> fromv1.0 -> programStructure.cpp
3. 加入 "p4\_logger",



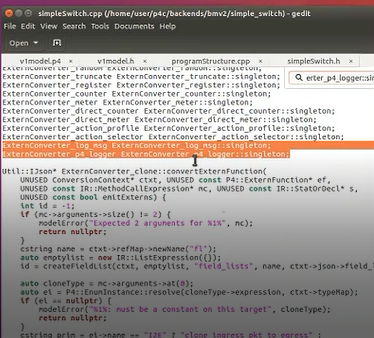
1. 繼續加下一個。Open -> other documents
2. user -> p4c -> backends -> bmv2 -> simple\_switch -> simpleSwitch.h
3. 加入EXTERN\_CONVERTER\_W\_FUNCTION(p4\_logger)



1. 繼續加下一個。Open -> other documents
2. user -> p4c -> backends -> bmv2 -> simple\_switch -> simpleSwitch.cpp

這個地方照抄網站上會錯！要照老師的改！

1. 加入ExternConverter\_p4\_logger ExternConverter\_p4\_logger::singleton;



1. 同檔案還有另一個地方要加，加入整段：

Util::IJson\* ExternConverter\_p4\_logger::convertExternFunction(

ConversionContext\* ctxt, UNUSED const P4::ExternFunction\* ef,

const IR::MethodCallExpression\* mc, const IR::StatOrDecl\* s,

UNUSED const bool emitExterns) {

if (mc->arguments->size() != 1)

{

modelError("Expected 1 arguments for %1%", mc);

return nullptr;

}

auto primitive = mkPrimitive("p4\_logger");

auto params = mkParameters(primitive);

primitive->emplace\_non\_null("source\_info", mc->sourceInfoJsonObj());

auto dest = ctxt->conv->convert(mc->arguments->at(0)->expression);

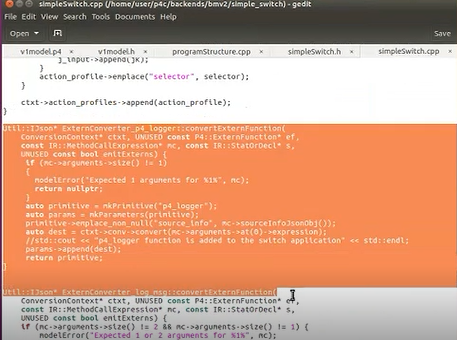
//std::cout << "p4\_logger function is added to the switch application" << std::endl;

params->append(dest);

return primitive;

}

整段插在Util::IJson\* ExternConverter\_log\_msg::convertExternFunction(上面



1. 最後一個加入。Open -> other documents
2. user -> behavior-model -> targets -> simple\_switch -> primitives.cpp
3. 加入整段

class p4\_logger :

public ActionPrimitive<const Data &> {

void operator()(const Data &operand) {

std::stringstream stream;

stream << std::hex << operand.get\_uint64();

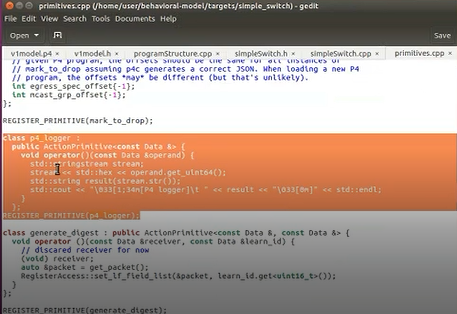
std::string result(stream.str());

std::cout << "\033[1;34m[P4 logger]\t " << result << "\033[0m]" << std::endl;

}

};

REGISTER\_PRIMITIVE(p4\_logger);



1. 做完以後，程式碼要重新編譯。把改好的程式碼save並關掉
2. 切到user/p4c/build資料夾
3. 執行make –j4
4. 執行make install
5. 跑完後切到user/ behavior-model /targets/simple\_switch資料夾
6. 執行make –j4
7. 執行make install

*沒有出錯就可以開始用了！*