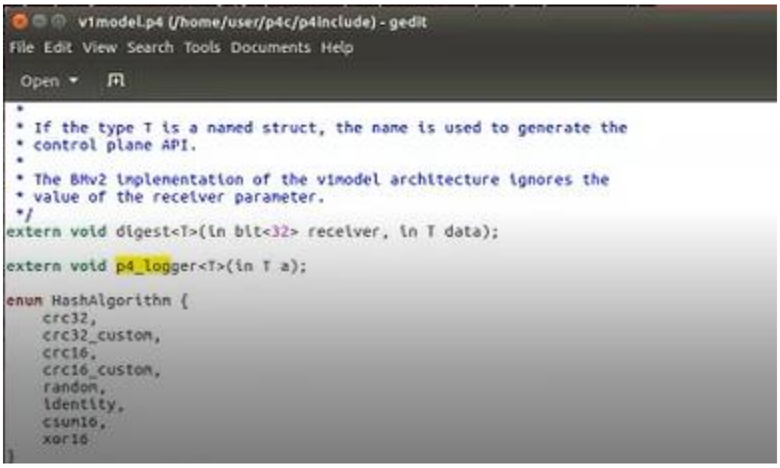
0615 在 p4 程式裡多一個函式：p4_logger

在開始程式的時候，想把那些東西印出來就可以印出來

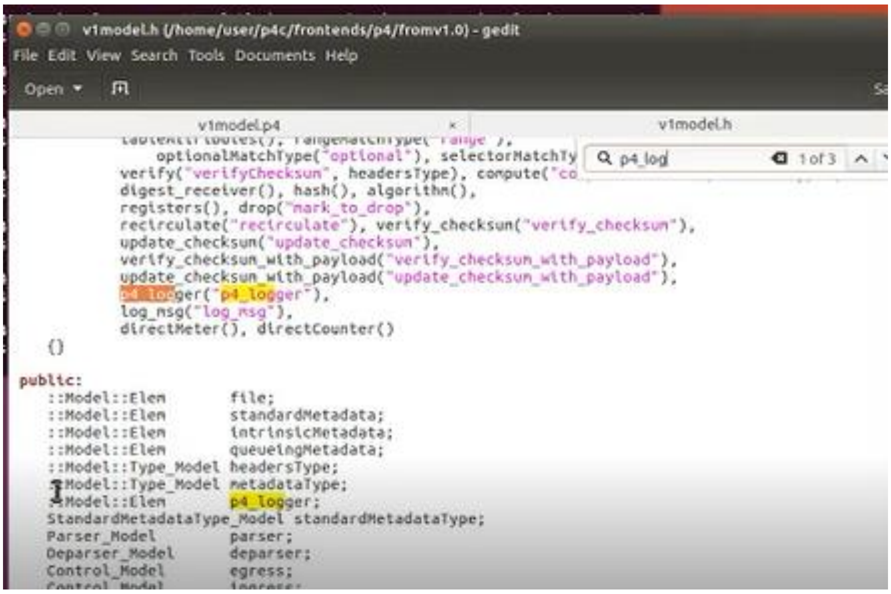
安裝步驟：

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


v1model.p4 (/home/user/p4c/p4include) - gedit
File Edit View Search Tools Documents Help
Open

```
 *
 * If the type T is a named struct, the name is used to generate the
 * control plane API.
 *
 * The BMv2 implementation of the v1model architecture ignores the
 * value of the receiver parameter.
 */
extern void digest<T>(in bit<32> receiver, in T data);

extern void p4_logger<T>(in T a);

enum HashAlgorithm {
    crc32,
    crc32_custom,
    crc16,
    crc16_custom,
    random,
    identity,
    csum16,
    xor16
}
```

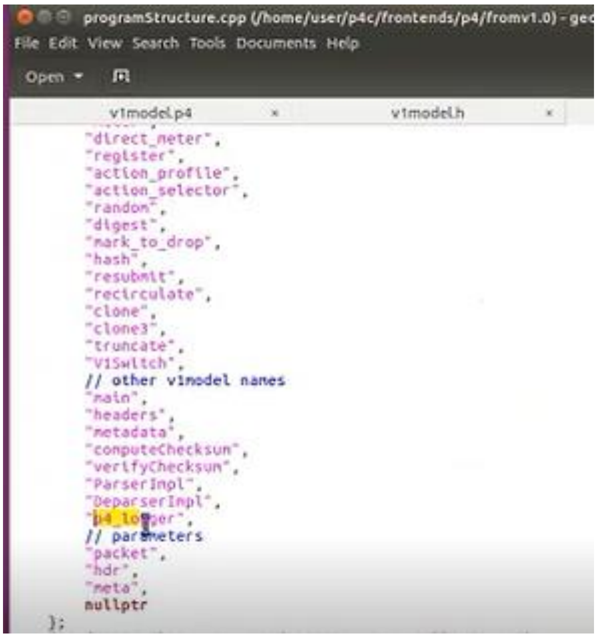
5. 加上以後，繼續加下一個。一樣 Open -> other documents
6. user -> p4c -> frontends -> p4 -> fromv1.0 -> v1model.h
7. 加入 p4_logger("p4_logger"), / ::Model::Elem p4_logger;


v1model.h (/home/user/p4c/frontends/p4/fromv1.0) - gedit
File Edit View Search Tools Documents Help
Open
v1model.p4
v1model.h
p4_log
1 of 3

```
        optionalMatchType("optional"), selectorMatchTy
    verify("verifyChecksum", headersType), compute("co
    digest_receiver(), hash(), algorithm(),
    registers(), drop("mark_to_drop"),
    recirculate("recirculate"), verify_checksum("verify_checksum"),
    update_checksum("update_checksum"),
    verify_checksum_with_payload("verify_checksum_with_payload"),
    update_checksum_with_payload("update_checksum_with_payload"),
    p4_logger("p4_logger"),
    log_msg("log_msg"),
    directMeter(), directCounter()
 {}

public:
    ::Model::Elem        file;
    ::Model::Elem        standardMetadata;
    ::Model::Elem        intrinsicMetadata;
    ::Model::Elem        queueingMetadata;
    ::Model::Type_Model  headersType;
    ::Model::Type_Model  metadataType;
    ::Model::Elem        p4_logger;
    StandardMetadataType_Model standardMetadataType;
    Parser_Model         parser;
    Deparser_Model       deparser;
    Control_Model        egress;
```

8. 繼續加下一個。Open -> other documents
9. user -> p4c -> frontends -> p4 -> fromv1.0 -> programStructure.cpp

10. 加入　"p4_logger",

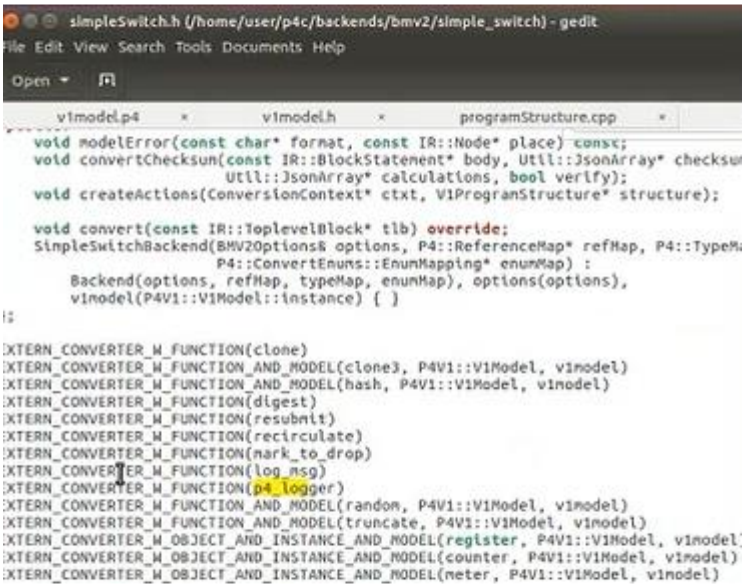

programStructure.cpp (/home/user/p4c/frontends/p4/fromv1.0) - ged
File Edit View Search Tools Documents Help
Open
v1model.p4
v1model.h

```
        "direct_meter",
        "register",
        "action_profile",
        "action_selector",
        "random",
        "digest",
        "mark_to_drop",
        "hash",
        "resubmit",
        "recirculate",
        "clone",
        "clone3",
        "truncate",
        "V1Switch",
        // other v1model names
        "main",
        "headers",
        "metadata",
        "computeChecksum",
        "verifyChecksum",
        "ParserImpl",
        "DeparserImpl",
        "p4_logger",
        // parameters
        "packet",
        "hdr",
        "meta",
        nullptr
    };
```

11. 繼續加下一個。Open -> other documents

12. user -> p4c -> backends -> bmv2 -> simple_switch -> simpleSwitch.h

13. 加入 EXTERN_CONVERTER_W_FUNCTION(p4_logger)


simpleSwitch.h (/home/user/p4c/backends/bmv2/simple_switch) - gedit
ile Edit View Search Tools Documents Help
Open
v1model.p4
v1model.h
programStructure.cpp

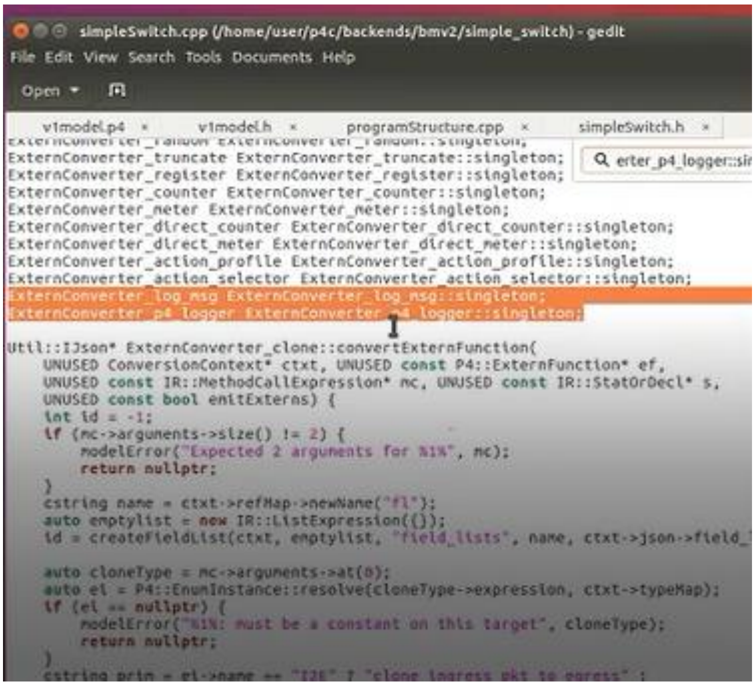
```
    void modelError(const char* format, const IR::Node* place) const;
    void convertChecksum(const IR::BlockStatement* body, Util::JsonArray* checksu
                         Util::JsonArray* calculations, bool verify);
    void createActions(ConversionContext* ctxt, V1ProgramStructure* structure);

    void convert(const IR::ToplevelBlock* tlb) override;
    SimpleSwitchBackend(BMV2Options& options, P4::ReferenceMap* refMap, P4::TypeMa
                        P4::ConvertEnums::EnumMapping* enumMap) :
        Backend(options, refMap, typeMap, enumMap), options(options),
        v1model(P4V1::V1Model::instance) { }
};

XTERN_CONVERTER_W_FUNCTION(clone)
XTERN_CONVERTER_W_FUNCTION_AND_MODEL(clone3, P4V1::V1Model, v1model)
XTERN_CONVERTER_W_FUNCTION_AND_MODEL(hash, P4V1::V1Model, v1model)
XTERN_CONVERTER_W_FUNCTION(digest)
XTERN_CONVERTER_W_FUNCTION(resubmit)
XTERN_CONVERTER_W_FUNCTION(recirculate)
XTERN_CONVERTER_W_FUNCTION(mark_to_drop)
XTERN_CONVERTER_W_FUNCTION(log_msg)
XTERN_CONVERTER_W_FUNCTION(p4_logger)
XTERN_CONVERTER_W_FUNCTION_AND_MODEL(random, P4V1::V1Model, v1model)
XTERN_CONVERTER_W_FUNCTION_AND_MODEL(truncate, P4V1::V1Model, v1model)
XTERN_CONVERTER_W_OBJECT_AND_INSTANCE_AND_MODEL(register, P4V1::V1Model, v1model
XTERN_CONVERTER_W_OBJECT_AND_INSTANCE_AND_MODEL(counter, P4V1::V1Model, v1model)
XTERN_CONVERTER_W_OBJECT_AND_INSTANCE_AND_MODEL(meter, P4V1::V1Model, v1model)
```

14. 繼續加下一個。Open -> other documents

15. 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！

16. 加入 ExternConverter_p4_logger ExternConverter_p4_logger::singleton;

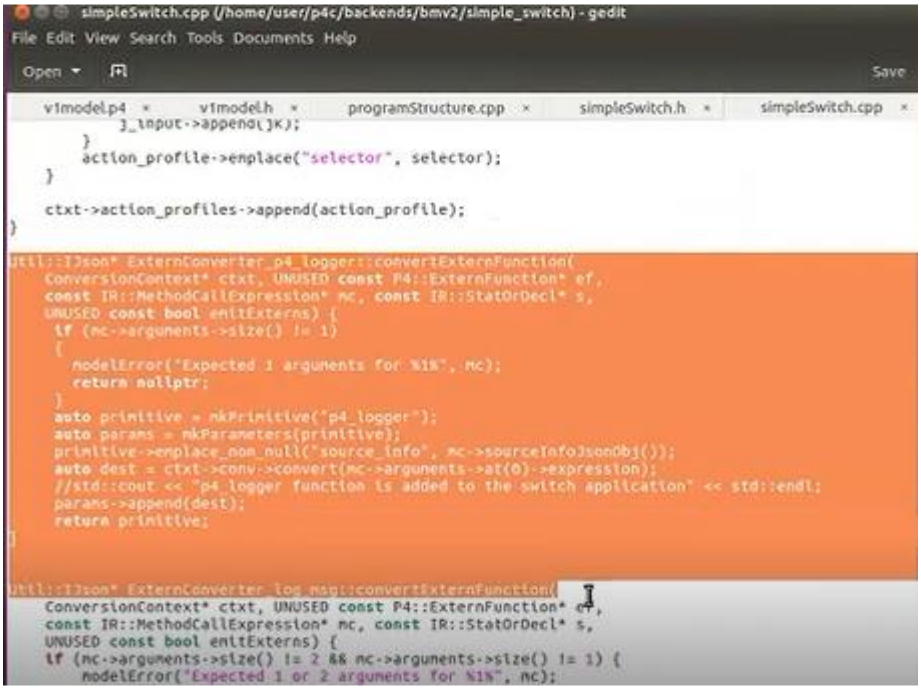

simpleSwitch.cpp (/home/user/p4c/backends/bmv2/simple_switch) - gedit
File Edit View Search Tools Documents Help
Open
v1model.p4
v1model.h
programStructure.cpp
simpleSwitch.h
erter_p4_logger::si

```
ExternConverter_random ExternConverter_random::singleton;
ExternConverter_truncate ExternConverter_truncate::singleton;
ExternConverter_register ExternConverter_register::singleton;
ExternConverter_counter ExternConverter_counter::singleton;
ExternConverter_meter ExternConverter_meter::singleton;
ExternConverter_direct_counter ExternConverter_direct_counter::singleton;
ExternConverter_direct_meter ExternConverter_direct_meter::singleton;
ExternConverter_action_profile ExternConverter_action_profile::singleton;
ExternConverter_action_selector ExternConverter_action_selector::singleton;
ExternConverter_log_msg ExternConverter_log_msg::singleton;
ExternConverter_p4_logger ExternConverter_p4_logger::singleton;

Util::IJson* ExternConverter_clone::convertExternFunction(
    UNUSED ConversionContext* ctxt, UNUSED const P4::ExternFunction* ef,
    UNUSED const IR::MethodCallExpression* mc, UNUSED const IR::StatOrDecl* s,
    UNUSED const bool emitExterns) {
    int id = -1;
    if (mc->arguments->size() != 2) {
        modelError("Expected 2 arguments for %1%", mc);
        return nullptr;
    }
    cstring name = ctxt->refMap->newName("fl");
    auto emptylist = new IR::ListExpression({});
    id = createFieldList(ctxt, emptylist, "field_lists", name, ctxt->json->field_

    auto cloneType = mc->arguments->at(0);
    auto ei = P4::EnumInstance::resolve(cloneType->expression, ctxt->typeMap);
    if (ei == nullptr) {
        modelError("%1%: must be a constant on this target", cloneType);
        return nullptr;
    }
    cstring prim = ei->name == "I2E" ? "clone_ingress_pkt_to_egress" :
```

17. 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;
}
```

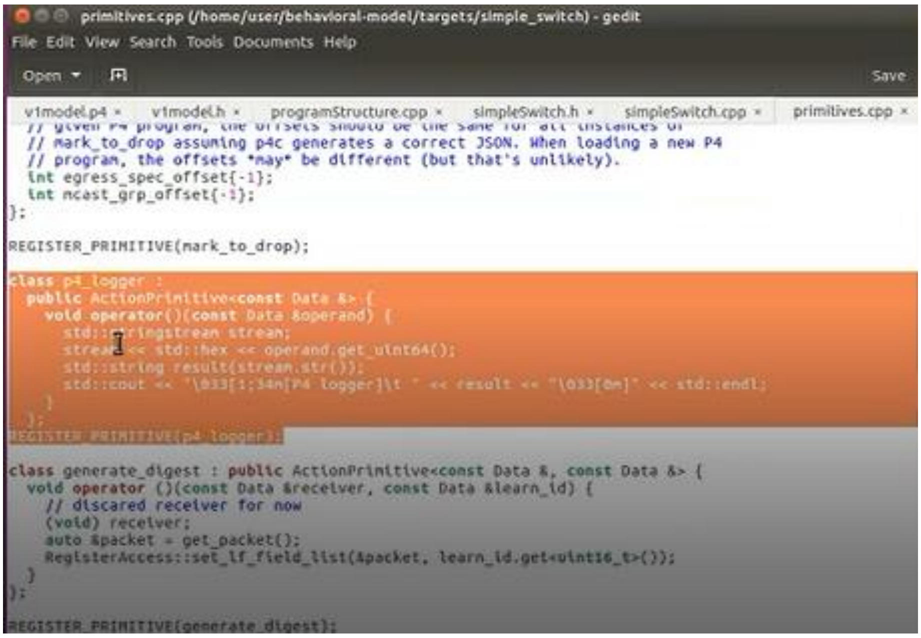
整段插在 Util::IJson* ExternConverter_log_msg::convertExternFunction(上面


simpleSwitch.cpp (/home/user/p4c/backends/bmv2/simple_switch) - gedit
File Edit View Search Tools Documents Help
Open
Save
v1model.p4
v1model.h
programStructure.cpp
simpleSwitch.h
simpleSwitch.cpp

```
            j_input->append(jk);
        }
        action_profile->emplace("selector", selector);
    }

    ctxt->action_profiles->append(action_profile);
}

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;
}

Util::IJson* ExternConverter_log_msg::convertExternFunction(
    ConversionContext* ctxt, UNUSED const P4::ExternFunction* ef,
    const IR::MethodCallExpression* mc, const IR::StatOrDecl* s,
    UNUSED const bool emitExterns) {
    if (mc->arguments->size() != 2 && mc->arguments->size() != 1) {
        modelError("Expected 1 or 2 arguments for %1%", mc);
```

18. 最後一個加入。Open -> other documents
19. user -> behavior-model -> targets -> simple_switch -> primitives.cpp
20. 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);
```



primitives.cpp (/home/user/behavioral-model/targets/simple_switch) - gedit
File Edit View Search Tools Documents Help
Open
Save
v1model.p4
v1model.h
programStructure.cpp
simpleSwitch.h
simpleSwitch.cpp
primitives.cpp

```
  // given P4 program, the offsets should be the same for all instances of
  // mark_to_drop assuming p4c generates a correct JSON. When loading a new P4
  // program, the offsets *may* be different (but that's unlikely).
  int egress_spec_offset{-1};
  int mcast_grp_offset{-1};
};

REGISTER_PRIMITIVE(mark_to_drop);

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);

class generate_digest : public ActionPrimitive<const Data &, const Data &> {
  void operator ()(const Data &receiver, const Data &learn_id) {
    // discared receiver for now
    (void) receiver;
    auto &packet = get_packet();
    RegisterAccess::set_lf_field_list(&packet, learn_id.get<uint16_t>());
  }
};

REGISTER_PRIMITIVE(generate_digest);
```

21. 做完以後，程式碼要重新編譯。把改好的程式碼 save 並關掉
22. 切到 user/p4c/build 資料夾
23. 執行 make –j4
24. 執行 make install
25. 跑完後切到 user/ behavior-model /targets/simple_switch 資料夾
26. 執行 make –j4
27. 執行 make install

    沒有出錯就可以開始用了！