

Разработка BSP для оптического коммутатора на базе Intel Tofino

Студент группы 8310
Слепов Артем

Постановка задачи

Цель работы - провести рефакторинг, добавить новый функционал и организовать систему сборки и доставки ПО BSP коммутатора-байпаса на основе Intel Tofino, а также разработать ПО IPMI-контроллера агрегатора-балансировщика.

Задачи работы:

- Проведение рефакторинга кода BSP для 16-портового коммутатора-байпаса, разделение функций по отдельным программным модулям.
- Подготовка системы сборки и доставки ПО, сборка собственного образа Linux с BSP 16-портового коммутатора-байпаса
- Уточнение требований и разработка BSP для 32-портового агрегатора-балансировщика с IPMI-контроллером.

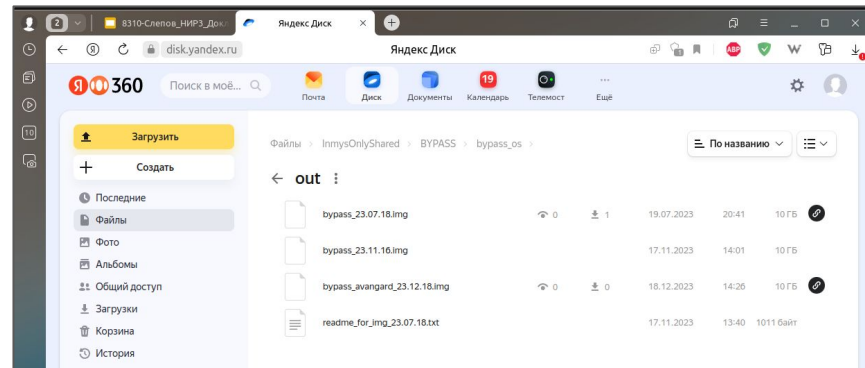
План работы на осенний семестр

1. Оформить набор разработанных скриптов в единый программный BSP на языке C++.
2. Подготовить инфраструктуру сборки, хранения и доставки ПО для платформы.

Статус



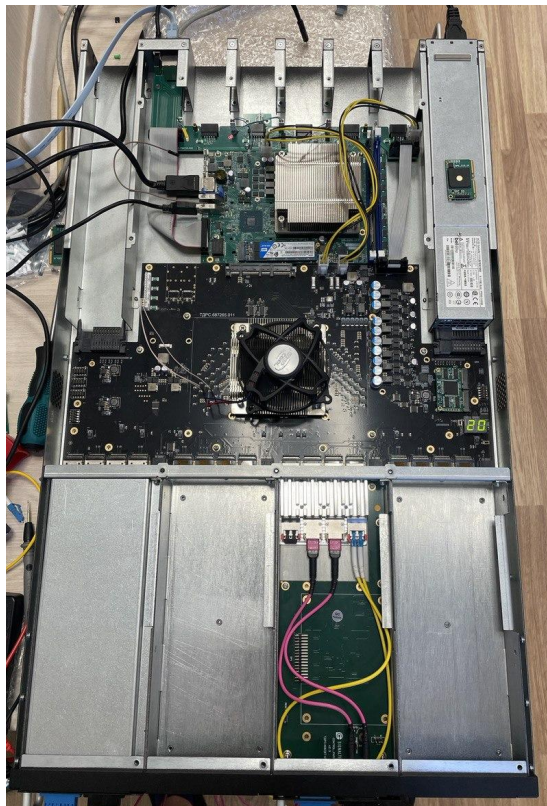
```
root@bypass:~# help
Tofino Test Suite
List of commands:
* bmc_version - read BMC FW version
* spm_ctl - all about CONTROLPASS: read-factory, ADC, power and OBP managment
* bmc_l2c_set/bmc_l2c_get - get/set BMC reg
* tofino_nms_serial - get TOFINO INMYS serial number
* ocp_nms_serial - get OCP INMYS serial number
* dump_qsfm PORTNUM - get QSFm information
BURN Suites:
* bmc_burn - burn bmc by $BMC_FW=/bypass-binaries/2023.06.02/stn32_4_3.hex firmware
* tofino_burn - burn tofino by $TOFINO_FW=/bypass-binaries/2023.06.02/tofino_B0_spt_gen2_rev06.bin firmware
* ispm_burn_fw - burn controlpass fw by module ID
* bios_burn - burn bios by $BIOS_BIN=/bypass-binaries/2023.06.02/coreboot.rom
Test Suites:
* qsfm_stm8_tester - test MEC connectors. Need QSFm_TESTER_V2 device
root@bypass:~# spm_c
-bash: spm_c: command not found
root@bypass:~# spm_ctl
spm_ctl help
usage: spm_ctl ACTION ARGUMENT
List of ACTION:
* enable - enable power controlpass module with ID=ARGUMENT[1,2,3,4,all]
* disable - disable power controlpass module with ID=ARGUMENT[1,2,3,4,all]
* obp - turn mirror in module with ID=ARG1 CHANNEL=ARG2 MODE=ARG3: CHANNEL=[0, 1]; MODE=[bypass, inline]
* read - read factory data and ADC measuring by module ID=ARG1
* monitor - read ADC measuring by module ID=ARG1
* reset - reset module with ID=[1,2,3,4] VAL=[0, 1]
* idled - led power on controlpass with ID=[1,2,3,4] VAL=[0, 1]
* boot0 - set boot0 pin in module with ID=[1,2,3,4] VAL=[0, 1]
* boot1 - set boot1 pin in module with ID=[1,2,3,4] VAL=[0, 1]
* serial - read serial number and module type with ID=[1,2,3,4]
root@bypass:~#
```



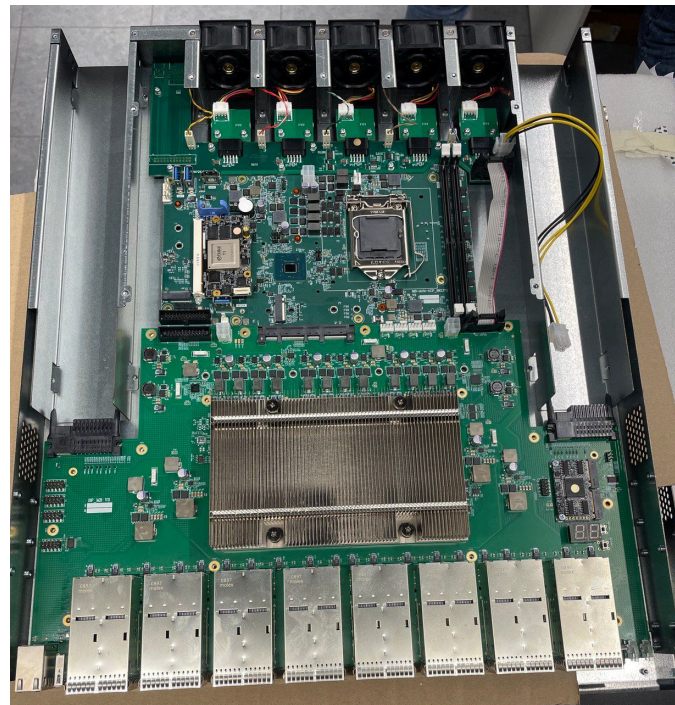
Релизы сборок ОС

Вид Command Line Interface после рефакторинга

Сравнение аппаратных платформ



Аппаратная платформа
коммутатора-байпаса



Аппаратная платформа
агрегатора-балансировщика

Практическая ценность разработки

Разработанное BSP и сборка Linux применяются:

- На производстве платформ коммутаторов-байпасов BYPASS SP100G4M в Санкт-Петербурге
- На производстве линейки из 10 модулей оптических переключателей BYPASS SPM в г. Долгопрудный
- В сервисном центре в г. Москва, который занимается диагностикой девайсов пришедших из эксплуатации в дата-центрах
- В лаборатории завода-разработчика, который занимается импортозамещением серверного блока питания PFE600-12-054NA

Секция ответов на вопросы