

Интеграция подсистемы запуска программ на исполнительской системе LuNA-6 в AZ



Руководитель: **Перепёлкин В.А.**

Студент: **Кудюков М.Н.**

Лаборатория синтеза параллельных программ, ИВМиМГ СО РАН
Летняя школа-конференция по параллельному программированию, 10 июля 2026 г.
<https://ssd.ssc.ru/ru/school/2026s>

Актуальность

- Концепция активных знаний позволяет автоматически применять накопленные программные модули для решения новых задач, снижая трудозатраты специалистов. Система LuNA-6 реализует эту концепцию, со временем система расширилась до автоматической генерации алгоритма в виде AZ LuNA-7, концептуально в AZ генерируется алгоритм и передаётся на исполнение в LuNA-6

Цель

- Создать “мост” между AZ и LuNA-6

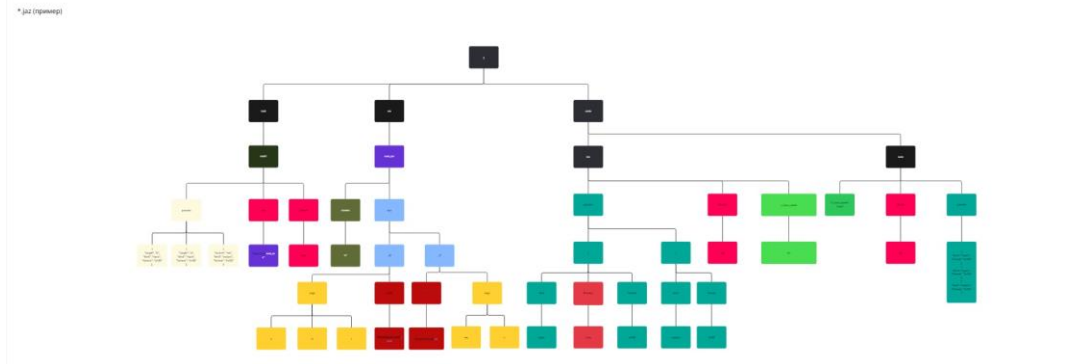
Задачи

1. Изучение свежего материала по AZ/LuNA-6
2. Реализовать скрипт конвертирующий свежий файл внутреннего представления AZ (*.jaz) в внутреннее представления LuNA-6 (*.ja)
3. Реализовать скрипт, который создаёт заготовку для программы, используя специальный промежуточный формат фрагмента кода.
4. Доработка запуска конвертирующих скриптов

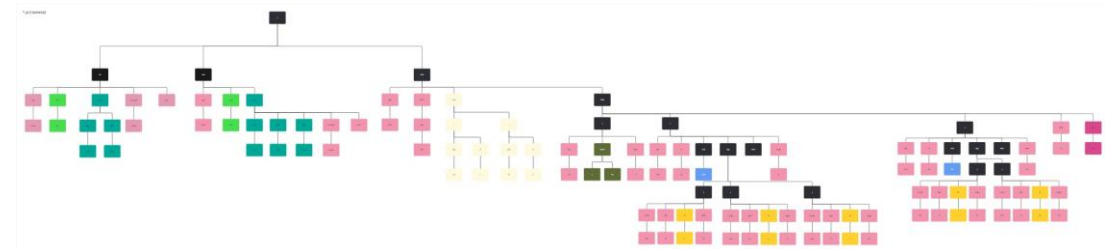
Задача 2:

Схемы структур json файлов AZ(*.jaz) и LuNA-6(*.ja) на примере: 04_sum_inc(из AZ)

AZ(*.jaz):



LuNA-6(*.ja):

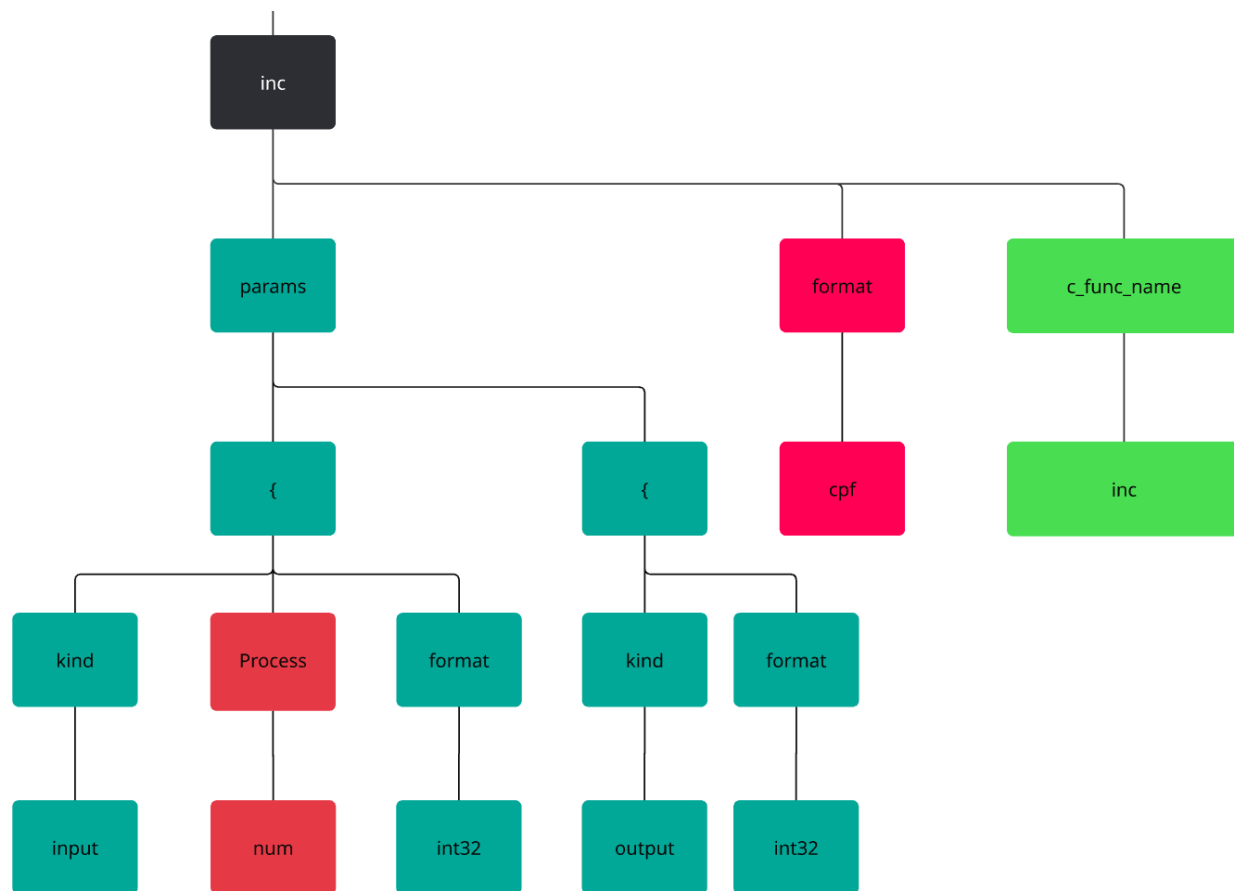


Ссылка на схему: https://miro.com/app/board/uXjVGKdS5Qo=/?share_link_id=11746607141 (не вечная)

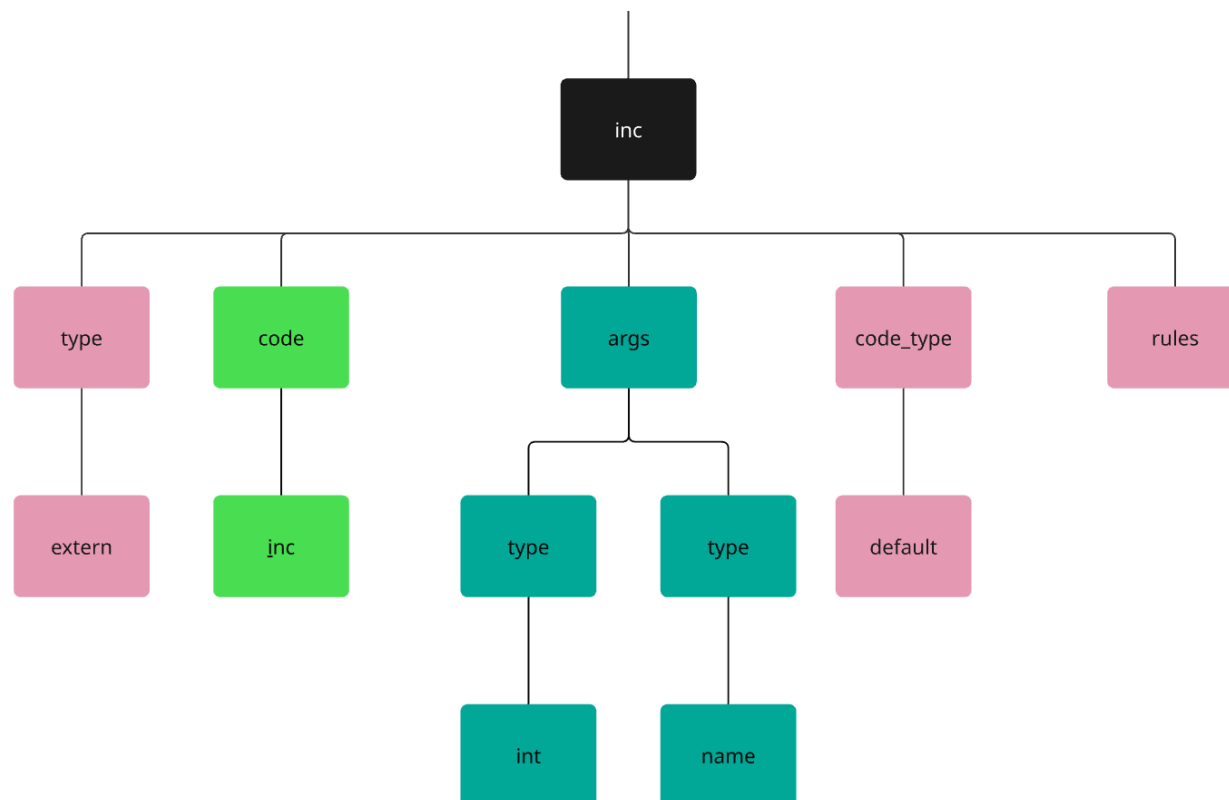
Задача 2:

Сравнимые части из структур AZ и LuNA-6

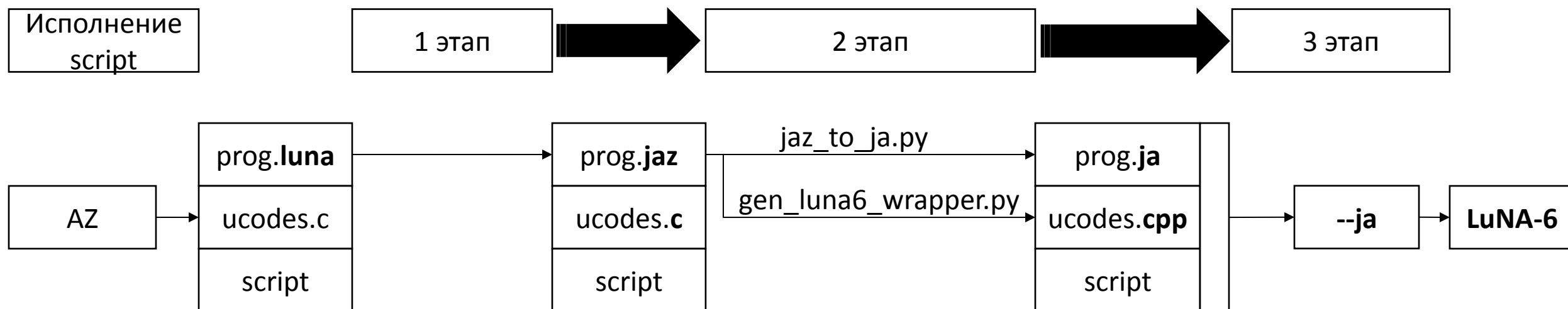
AZ(*.jaz): code: inc:



LuNA-6(*.ja): inc:



Пример работы скриптов задачи 2 и 3 и спецификатора 4 (схематично)



Примечание:

jaz_to_ja.py == 2 задача, скрипт конвертирующий

gen_luna6_wrapper.py == 3 задача, скрипт создающий оболочку

--ja == спецификатор в LuNA-6

Тестирование:

Статус запущенных примеров : Причины ошибок:

AZ(2025):

✓ 01_inc
✓ 03_real

✓ 04_sum_inc
✓ 05_sum_inc2
✓ 06_minmax
✓ 07_arr
✓ 08_loop
✓ 09_path
✓ 10_plan
✓ 11_string
✓ 13_corr
✓ 14_web_service
✗ 15_triangle_llm
✓ 16_clif_sh

AZ(2026):

✓ 01_inc
✓ 03_real
✗ 02_path
✗ 04_sum_inc
✗ 05_sum_inc2
✗ 06_minmax
✗ 07_arr
✗ 08_loop
✗ 09_path
✗ 10_plan
✗ 11_string
✗ 13_corr
✗ 14_web_service
✗ 15_triangle_llm
✗ 16_clif_sh

Примеры не работают за счёт
изменения внутреннего
представления az в самой последней
версии

Заключение

- Сделан прототип “моста” между AZ и LuNA-6
- В следующих версиях стоит цель покрытия всех примеров и обновления конвертеров для обработки последних типов *.jaz файлов

Спасибо за внимание!

