

# PROTEI UDM/UDR

Руководство по эксплуатации

# Содержание

Юридическая информация .....	1
1. Общее описание .....	2
1.1. Общие сведения .....	2
1.2. Функциональные возможности .....	2
1.3. Сетевая архитектура .....	3
2. Компоненты системы .....	4
2.1. UDM .....	4
2.2. UDR .....	7
2.3. AUSF .....	10

## Юридическая информация

Информация, содержащаяся в настоящем документе, является конфиденциальной, предназначена исключительно для использования Заказчиком (Конечным пользователем) и не подлежит передаче третьим лицам. Любое воспроизведение, распространение и/или использование информации, содержащейся в настоящем документе, а также отдельных выдержек из нее, в том числе с изменениями или переводом на другие языки, осуществляется исключительно с предварительного письменного согласия, полученного от ООО "НТЦ ПРОТЕЙ".

ООО "НТЦ ПРОТЕЙ" обладает исключительными правами на настоящий документ и информацию, содержащуюся в нем, и оставляет за собой право вносить любые изменения в настоящий документ без предварительного уведомления Заказчика (Конечного пользователя).

ООО "НТЦ ПРОТЕЙ" не несет перед Заказчиком (Конечным пользователем) и/или третьими лицами ответственности за убытки (включая упущенную выгоду), которые могут возникнуть в результате применения Заказчиком (Конечным пользователем) и/или третьими лицами настоящего документа.

194044, Санкт-Петербург  
Большой Сампсониевский пр., д. 60, лит. А  
Бизнес-центр "Телеком"  
Тел.: (812) 449-47-27  
Факс: (812) 449-47-29  
Web: <https://www.protei.ru>  
Email: [info@protei.ru](mailto:info@protei.ru)

# 1. Общее описание

## 1.1. Общие сведения

Система PROTEI UDM/UDR представляет собой центральный защищенный узел, предназначенный для управления профилями абонентов, выполнения процедур аутентификации и авторизации.

Система PROTEI UDM/UDR реализована в сервисно-ориентированной архитектуре (SBA), имеет распределенную структуру и состоит из нескольких компонентов (сервисов), которые выполняют определенный набор функций и взаимодействуют друг с другом с помощью стандартизированных интерфейсов и протоколов. Такое решение обеспечивает независимость отдельных функциональных модулей, упрощает масштабирование и развитие сети.

Решение PROTEI UDM/UDR состоит из следующих компонентов:

- PROTEI UDM (Unified Data Management) — сервис управления данными абонентов, обеспечивающий обработку и хранение данных подписки, управление регистрацией абонентов в сети, предоставление данных сетевым функциям, а также поддержку процедур мобильности и управления доступом;
- PROTEI UDR (Unified Data Repository) — централизованное хранилище данных абонентов и сервисных профилей, обеспечивающее безопасное хранение, доступ и синхронизацию информации для сетевых функций 5G Core;
- PROTEI AUSF (Authentication Server Function) — сервис аутентификации абонентов, обеспечивающий выполнение процедур 5G AKA/EAP-AKA', взаимодействие с UDM и генерацию параметров аутентификации в соответствии с требованиями 3GPP.

UDM, UDR, AUSF используют протокол RESTful API для обмена данными с другими сетевыми элементами (NF 5G) по внешним интерфейсам (Nudm, Nudr, Nausf).

Подробнее о каждом компоненте системы приведено в разделах 2,3 и 4 настоящего документа.

Архитектура PROTEI UDM/UDR обеспечивает высокую отказоустойчивость, гибкость масштабирования и модернизации, защищенное хранение и обработку данных абонентов, а также упрощенную интеграцию с сетевыми функциями 5G Core в соответствии с требованиями 3GPP Release 17.

## 1.2. Функциональные возможности

Ключевые функции компонентов системы PROTEI UDM/UDR:

- Генерация векторов аутентификации;
- Управление данными подписки;
- Хранение профилей подписки и параметров услуг;
- Поддержка регистрации абонентов в сети 5G;
- Предоставление данных другим сетевым функциям 5G Core;

- Поддержка процедур авторизации и управления доступом;
- Обеспечение взаимодействия с сетевыми функциями по интерфейсам SBA;
- Хранение данных подписки для VoNR и сервисов передачи данных;
- Обеспечение защищенного хранения и обработки данных абонентов;
- Управление регистрацией обслуживающих NF;
- Подписка на уведомления об изменении данных (Data change notifications);
- Преобразование скрытого идентификатора (SUCI) в постоянный (SUPI).

### 1.3. Сетевая архитектура

Структурная схема системы PROTEI UDM/UDR и ее место в сетевой архитектуре приведены на рисунке 1.

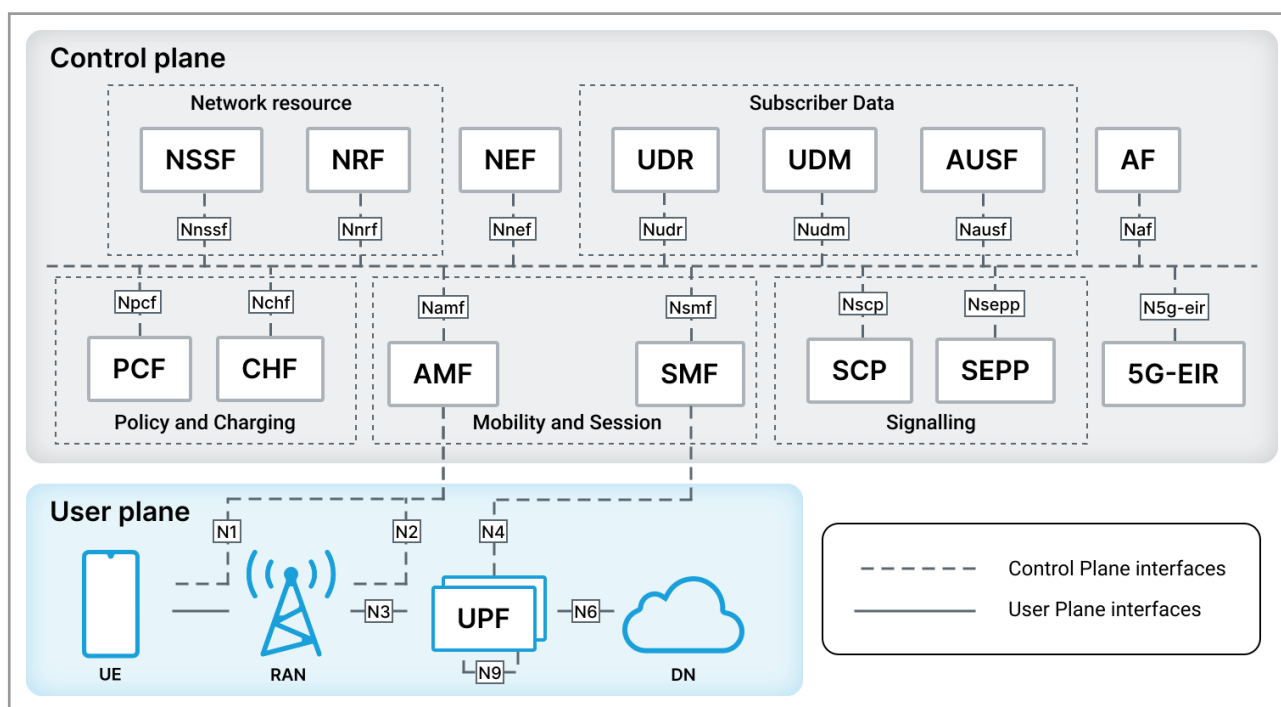


Рисунок 1. Схема сетевой архитектуры

## 2. Компоненты системы

### 2.1. UDM

PROTEI UDM (Unified Data Management) — сетевая функция в 5G Core, которая предназначена для централизованного управления данными пользователей и выполнения администрирования абонентов. UDM обеспечивает обработку, хранение и предоставление профилей пользователей, данных аутентификации и информации о подписке (subscription data) для других сетевых функций.

Узел PROTEI UDM выполняет следующие задачи:

- Хранение и обработка идентификаторов пользователя (SUPI, SUCI);
- Взаимодействие с другими сетевыми функциями;
- Управление подписками и параметрами услуг;
- Обработка данных о подписчиках;
- Управление профилями пользователей;
- Управление регистрацией абонента;
- Генерация векторов аутентификации (5G AKA);
- Управление регистрацией обслуживающих сетевых функций (NF);
- Подписка на уведомления об изменении данных.

UDM взаимодействует с AMF, SMF, AUSF, PCF и NEF для обмена идентификаторами SUPI/GPSI, данными аутентификации, профилями подписки и контекстом регистрации. В отличие от Home Subscriber Service (HSS) в сети 4G, управление данными выполняется через интерфейс Nudr к UDR, который выступает в роли постоянного хранилища.

Подробная информация по эксплуатации решения UDM (конфигурирование, API) представлена в [документации PROTEI UDM](#).

#### 2.1.1. Управление

##### Директории

В PROTEI UDM используются следующие директории:

- `/usr/protei/Protei_UDM/cdr/` — директория для журналов CDR;
- `/usr/protei/Protei_UDM/config/` — директория для конфигурационных файлов.

##### Управляющие команды

- Чтобы запустить PROTEI UDM, следует выполнить одну из команд:
- команду `systemctl start` от лица суперпользователя:

```
```bash
$ sudo systemctl start udm
```
```

- скрипт start в рабочей папке:

```
```bash
$ /usr/protei/Protei_UDM/start
```
```

- Чтобы остановить PROTEI UDM, следует выполнить одну из команд:
- команду `systemctl stop` от лица суперпользователя:

```
```bash
$ sudo systemctl stop udm
```
```

- скрипт stop в рабочей папке:

```
```bash
$ /usr/protei/Protei_UDM/stop
```
```

- Чтобы проверить текущее состояние PROTEI UDM, следует выполнить одну из команд:
- команду `systemctl status` от лица суперпользователя:

```
```bash
$ sudo systemctl status udm
● udm.service - UDM
   Loaded: loaded (/usr/lib/systemd/system/udm.service; enabled; vendor preset: disable>
   Drop-In: /etc/systemd/system/udm.service.d
            └─override.conf
   Active: active (running) since Fri 2023-07-21 19:50:25 MSK; 2 days ago
   Process: 998814 ExecStopPost=/usr/protei/Protei_UDM/bin/utils/check_history.sh
            (code=exited, status=0/SUCCESS)
   Process: 998729 ExecStopPost=/usr/protei/Protei_UDM/bin/utils/move_log.sh (code=exited,
            status=0/SUCCESS)
   Process: 998683 ExecStop=/usr/protei/Protei_UDM/bin/utils/stop_prog.sh (code=exited,
            status=0/SUCCESS)
   Process: 999135 ExecStart=/usr/protei/Protei_UDM/bin/utils/start_prog.sh (code=exited,
            status=0/SUCCESS)
   Process: 999108 ExecStartPre=/usr/protei/Protei_UDM/bin/utils/check_history.sh
            (code=exited, status=0/SUCCESS)
   Process: 999082 ExecStartPre=/usr/protei/Protei_UDM/bin/utils/move_log.sh (code=exited,
            status=0/SUCCESS)
   Main PID: 999151 (Protei_UDM)
      Tasks: 11 (limit: 35612)
     Memory: 489.7M
    CGroup: /system.slice/udm.service
            └─999151 ./bin/Protei_UDM
```
```

```
```
```

- скрипт `status` в рабочей папке:

```
```bash
$ /usr/protei/Protei_UDM/status
```
```

- Чтобы проверить текущую версию PROTEI UDM, следует запустить скрипт `version` в рабочей папке:

```
```bash
$ /usr/protei/Protei_UDM/version
```
```

- Чтобы перезагрузить PROTEI UDM, следует выполнить одну из команд:
- команду `systemctl restart` от лица суперпользователя:

```
```bash
$ sudo systemctl restart udm
● udm.service - UDM
   Loaded: loaded (/usr/lib/systemd/system/udm.service; enabled; vendor preset: disabled)
   Active: active (running) since Fri 2023-07-21 19:50:25 MSK; 2 days ago
     Process: 99426 ExecStopPost=/usr/protei/Protei_UDM/bin/utils/check_history.sh
              (code=exited, status=0/SUCCESS)
     Process: 99344 ExecStopPost=/usr/protei/Protei_UDM/bin/utils/move_log.sh (code=exited,
              status=0/SUCCESS)
     Process: 99300 ExecStop=/usr/protei/Protei_UDM/bin/utils/stop_prog.sh (code=exited,
              status=0/SUCCESS)
     Process: 99508 ExecStart=/usr/protei/Protei_UDM/bin/utils/start_prog.sh (code=exited,
              status=0/SUCCESS)
     Process: 99481 ExecStartPre=/usr/protei/Protei_UDM/bin/utils/check_history.sh
              (code=exited, status=0/SUCCESS)
     Process: 99453 ExecStartPre=/usr/protei/Protei_UDM/bin/utils/move_log.sh (code=exited,
              status=0/SUCCESS)
   Main PID: 99524 (Protei_UDM)
     Tasks: 1 (limit: 35612)
    Memory: 2.5M
    CGroup: /system.slice/udm.service
            └─99524 ./bin/Protei_UDM
```
```

- скрипт `restart` в рабочей папке:

```
```bash
$ /usr/protei/Protei_UDM/restart
```
```

- Чтобы записать дамп ядра, следует запустить скрипт `core_dump` в рабочей папке:

```
```bash
$ /usr/protei/Protei_UDM/core_dump
Are you sure you want to continue? y
Core dump generated!
```
```

**Примечание.** Файл дампа хранится в директории `/var/lib/systemd/coredump`.

- Чтобы перезагрузить конфигурационный файл `file.json`, следует запустить скрипт `reload` в рабочей папке:

```
```bash
$ /usr/protei/Protei_UDM/reload <file.json>
reload <file> config 0k
```
```

## 2.2. UDR

PROTEI UDR (Unified Data Repository) — ключевая сетевая функция архитектуры 5G Core (5GC), которая выполняет роль централизованного хранилища абонентских данных, данных подписки, политик и информации о тарификации.

UDR взаимодействует с provisioning-сервисами для актуализации абонентских профилей с помощью платформы управления абонентскими данными (SDM). Поддерживает горизонтальное масштабирование для обеспечения высокой доступности и взаимодействует с сетевыми функциями через интерфейсы Nudr.

Узел PROTEI UDR выполняет следующие функции:

- Подписка на уведомления об изменении данных;
- Управление данными подписки;
- Хранение и применение политик;
- Управление данными тарификации;
- Шифрование данных;
- Регистрация и обнаружение профиля через NRF, с возможностью маршрутизации через SCP;
- Взаимодействие с AUSF через узел UDM;
- Поддержка горизонтального масштабирования при увеличении количества абонентов и трафика;
- Обеспечение высокой доступности за счет балансировки нагрузки и наличия распределенных баз данных.

Подробная информация по эксплуатации решения UDR (конфигурирование, API) представлена в [документации PROTEI UDR](#).

## 2.2.1. Управление

### Директории

В PROTEI UDR используются следующие директории:

- /usr/protei/Protei\_UDR/ — рабочая директория;
- /usr/protei/Protei\_UDR/config/ — директория для конфигурационных файлов.

### Управляющие команды

- Чтобы запустить PROTEI UDR, следует выполнить одну из команд:

- команду `systemctl start` от лица суперпользователя:

```
$ sudo systemctl start udr
```

- скрипт `start` в рабочей папке:

```
$ /usr/protei/Protei_UDR/start
```

- Чтобы остановить PROTEI UDR, следует выполнить одну из команд:

- команду `systemctl stop` от лица суперпользователя:

```
$ sudo systemctl stop udr
```

- скрипт `stop` в рабочей папке:

```
$ /usr/protei/Protei_UDR/stop
```

- Чтобы проверить текущее состояние PROTEI UDR, следует выполнить одну из команд:

- команду `systemctl status` от лица суперпользователя:

```
$ sudo systemctl status udr
● UDR.service - UDR
   Loaded: loaded (/usr/lib/systemd/system/udr.service; enabled; vendor preset: disable>
   Drop-In: /etc/systemd/system/udr.service.d
            └─override.conf
   Active: active (running) since Fri 2023-07-21 19:50:25 MSK; 2 days ago
   Process: 998814 ExecStopPost=/usr/protei/Protei_UDR/bin/utills/check_history.sh
            (code=exited, status=0/SUCCESS)
   Process: 998729 ExecStopPost=/usr/protei/Protei_UDR/bin/utills/move_log.sh
            (code=exited, status=0/SUCCESS)
   Process: 998683 ExecStop=/usr/protei/Protei_UDR/bin/utills/stop_prog.sh (code=exited,
            status=0/SUCCESS)
   Process: 999135 ExecStart=/usr/protei/Protei_UDR/bin/utills/start_prog.sh (code=exited,
            status=0/SUCCESS)
   Process: 999108 ExecStartPre=/usr/protei/Protei_UDR/bin/utills/check_history.sh
```

```
(code=exited, status=0/SUCCESS)
Process: 999082 ExecStartPre=/usr/protei/Protei_UDR/bin/utils/move_log.sh
(code=exited, status=0/SUCCESS)
Main PID: 999151 (Protei_UDR)
Tasks: 11 (limit: 35612)
Memory: 489.7M
CGroup: /system.slice/udr.service
└─999151 ./bin/Protei_UDR
```

- скрипт `status` в рабочей папке:

```
$ /usr/protei/Protei_UDR/status
```

- Чтобы проверить текущую версию PROTEI UDR, следует запустить скрипт `version` в рабочей папке:

```
$ /usr/protei/Protei_UDR/version
```

- Чтобы перезагрузить PROTEI UDR, следует выполнить одну из команд:

- команду `systemctl restart` от лица суперпользователя:

```
$ sudo systemctl restart udr
● UDR.service - UDR
   Loaded: loaded (/usr/lib/systemd/system/udr.service; enabled; vendor preset: disabled)
   Active: active (running) since Fri 2023-07-21 19:50:25 MSK; 2 days ago
     Process: 99426 ExecStopPost=/usr/protei/Protei_UDR/bin/utils/check_history.sh (code=exited, status=0/SUCCESS)
     Process: 99344 ExecStopPost=/usr/protei/Protei_UDR/bin/utils/move_log.sh (code=exited, status=0/SUCCESS)
     Process: 99300 ExecStop=/usr/protei/Protei_UDR/bin/utils/stop_prog.sh (code=exited, status=0/SUCCESS)
     Process: 99508 ExecStart=/usr/protei/Protei_UDR/bin/utils/start_prog.sh (code=exited, status=0/SUCCESS)
     Process: 99481 ExecStartPre=/usr/protei/Protei_UDR/bin/utils/check_history.sh (code=exited, status=0/SUCCESS)
     Process: 99453 ExecStartPre=/usr/protei/Protei_UDR/bin/utils/move_log.sh (code=exited, status=0/SUCCESS)
  Main PID: 99524 (Protei_UDR)
    Tasks: 1 (limit: 35612)
   Memory: 2.5M
   CGroup: /system.slice/udr.service
           └─99524 ./bin/Protei_UDR
```

- скрипт `restart` в рабочей папке:

```
$ /usr/protei/Protei_UDR/restart
```

- Чтобы перезагрузить конфигурационный файл `file.json` и `file.cfg`, следует запустить скрипт `reload` в рабочей папке:

```
$ /usr/protei/Protei_UDR/reload <file.json>
reload <file> config Ok
```

- Чтобы записать дампы ядра, следует запустить скрипт `core_dump` в рабочей папке:

```
$ /usr/protei/Protei_UDR/core_dump
Are you sure you want to continue? y
Core dump generated!
```

**Примечание.** Файл дампа хранится в директории `/var/lib/systemd/coredump`.

## 2.3. AUSF

PROTEI AUSF (Authentication Server Function) — ключевой элемент базовой сети 5G, предназначенный для аутентификации пользовательского оборудования (UE) в домашней сети. AUSF принимает решение об успешности аутентификации, взаимодействуя с сетевой функцией UDM для получения необходимых данных и ключей. Поддерживает современные методы аутентификации, такие как 5G-AKA и EAP-AKA', обеспечивая безопасный доступ к услугам сети.

Основные функции узла PROTEI AUSF:

- Взаимодействие с UDM и UDR для получения данных и ключей, необходимых для аутентификации;
- Поддержка современных методов аутентификации:
  - 5G-AKA;
  - EAP-AKA';
- Обеспечение безопасного доступа к услугам сети;
- Контроль выполнения аутентификации;
- Получение постоянного идентификатора абонента (SUPI) от UDM.

Совместно с узлами UDM (Unified Data Management) и UDR (Unified Data Repository) AUSF является частью архитектуры 5G Core, реализующей механизмы аутентификации в соответствии с 3GPP Release 17 и обеспечивающей функции, необходимые для предоставления услуг передачи данных и голосовой связи по сети 5G (VoNR).

Подробная информация по эксплуатации решения AUSF (конфигурирование, API) представлена в [документации PROTEI AUSF](#).

### 2.3.1. Управление

#### Директории

В PROTEI AUSF используются следующие директории:

- /usr/protei/Protei\_AUSF/ — рабочая директория;
- /usr/protei/Protei\_AUSF/config/ — директория для конфигурационных файлов.

### Управляющие команды

- Чтобы запустить PROTEI AUSF, следует выполнить одну из команд:
  - команду `systemctl start` от лица суперпользователя:

```
$ sudo systemctl start ausf
```

- скрипт `start` в рабочей папке:

```
$ /usr/protei/Protei_AUSF/start
```

- Чтобы остановить PROTEI AUSF, следует выполнить одну из команд:
  - команду `systemctl stop` от лица суперпользователя:

```
$ sudo systemctl stop ausf
```

- скрипт `stop` в рабочей папке:

```
$ /usr/protei/Protei_AUSF/stop
```

- Чтобы проверить текущее состояние PROTEI AUSF, следует выполнить одну из команд:
  - команду `systemctl status` от лица суперпользователя:

```
$ sudo systemctl status ausf
● ausf.service - AUSF
   Loaded: loaded (/usr/lib/systemd/system/ausf.service; enabled; vendor preset:
  disable>
   Drop-In: /etc/systemd/system/ausf.service.d
            └─override.conf
   Active: active (running) since Fri 2023-07-21 19:50:25 MSK; 2 days ago
   Process: 998814 ExecStopPost=/usr/protei/Protei_AUSF/bin/utls/check_history.sh
  (code=exited, status=0/SUCCESS)
   Process: 998729 ExecStopPost=/usr/protei/Protei_AUSF/bin/utls/move_log.sh
  (code=exited, status=0/SUCCESS)
   Process: 998683 ExecStop=/usr/protei/Protei_AUSF/bin/utls/stop_prog.sh (code=exited,
  status=0/SUCCESS)
   Process: 999135 ExecStart=/usr/protei/Protei_AUSF/bin/utls/start_prog.sh
  (code=exited, status=0/SUCCESS)
   Process: 999108 ExecStartPre=/usr/protei/Protei_AUSF/bin/utls/check_history.sh
  (code=exited, status=0/SUCCESS)
   Process: 999082 ExecStartPre=/usr/protei/Protei_AUSF/bin/utls/move_log.sh
  (code=exited, status=0/SUCCESS)
  Main PID: 999151 (Protei_AUSF)
     Tasks: 11 (limit: 35612)
    Memory: 489.7M
    CGroup: /system.slice/ausf.service
```

```
└─999151 ./bin/Protei_AUSF
```

- скрипт status в рабочей папке:

```
$ /usr/protei/Protei_AUSF/status
```

- Чтобы проверить текущую версию PROTEI AUSF, следует запустить скрипт version в рабочей папке:

```
$ /usr/protei/Protei_AUSF/version
```

- Чтобы перезагрузить PROTEI AUSF, следует выполнить одну из команд:

- команду `systemctl restart` от лица суперпользователя:

```
$ sudo systemctl restart ausf
● ausf.service - AUSF
   Loaded: loaded (/usr/lib/systemd/system/ausf.service; enabled; vendor preset: disabled)
   Active: active (running) since Fri 2023-07-21 19:50:25 MSK; 2 days ago
     Process: 99426 ExecStopPost=/usr/protei/Protei_AUSF/bin/utills/check_history.sh (code=exited, status=0/SUCCESS)
     Process: 99344 ExecStopPost=/usr/protei/Protei_AUSF/bin/utills/move_log.sh (code=exited, status=0/SUCCESS)
     Process: 99300 ExecStop=/usr/protei/Protei_AUSF/bin/utills/stop_prog.sh (code=exited, status=0/SUCCESS)
     Process: 99508 ExecStart=/usr/protei/Protei_AUSF/bin/utills/start_prog.sh (code=exited, status=0/SUCCESS)
     Process: 99481 ExecStartPre=/usr/protei/Protei_AUSF/bin/utills/check_history.sh (code=exited, status=0/SUCCESS)
     Process: 99453 ExecStartPre=/usr/protei/Protei_AUSF/bin/utills/move_log.sh (code=exited, status=0/SUCCESS)
    Main PID: 99524 (Protei_AUSF)
      Tasks: 1 (limit: 35612)
     Memory: 2.5M
    CGroup: /system.slice/ausf.service
            └─99524 ./bin/Protei_AUSF
```

- скрипт restart в рабочей папке:

```
$ /usr/protei/Protei_AUSF/restart
```

- Чтобы перезагрузить конфигурационный файл `file.json` и `file.cfg`, следует запустить скрипт `reload` в рабочей папке:

```
$ /usr/protei/Protei_AUSF/reload <file.json>
reload <file> config 0k
```

- Чтобы записать дампы ядра, следует запустить скрипт `core_dump` в рабочей папке:

```
$ /usr/protei/Protei_AUSF/core_dump  
Are you sure you want to continue? y  
Core dump generated!
```

**Примечание.** Файл дампа хранится в директории `/var/lib/systemd/coredump`.