

ООО «НПО «Аквастандарт»

**Установка программного обеспечения
«ГМП-26» для ОС Linux**

Санкт-Петербург 2020

Содержание

1 Технические требования к ПО ГМП-26	3
2 Установка программного обеспечения	4
3 Запуск программного обеспечения	4

1 Основные технические требования к машине, на который устанавливается программное обеспечение «ГМП-26»

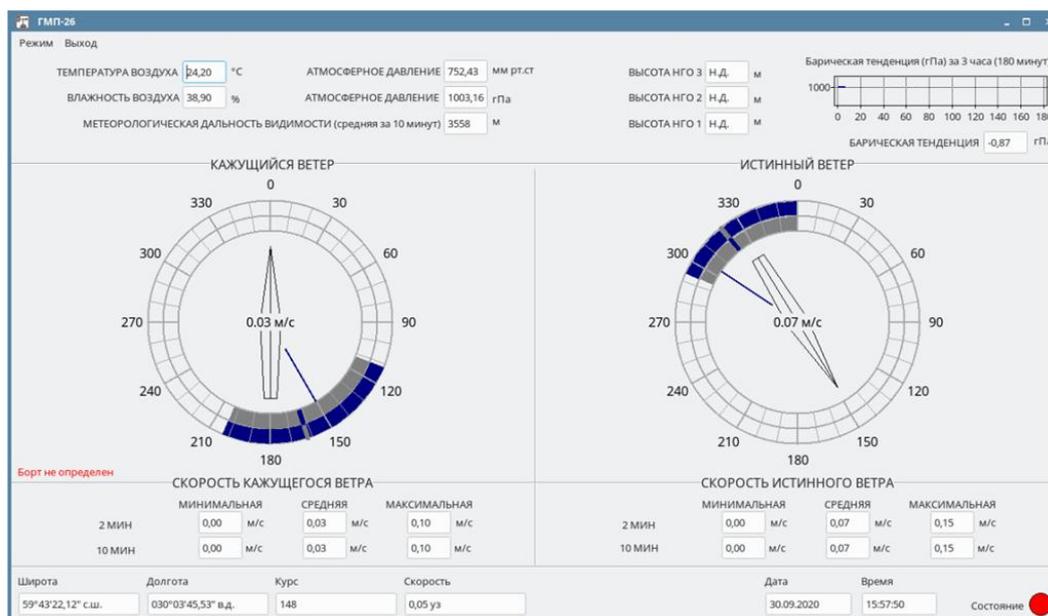


Рисунок 1 — внешний вид главного окна

Программное обеспечение далее (ПО «ГМП-26») предназначено для приема, обработки и отображения гидрометеорологической информации, полученной от различных финских датчиков: WMT703, PWD50, РТВ330, CL31, а также отечественных метеорологических датчиков ДСНВАЦ-8, ДНГО-8, ДТВЦ-4, ДАДЦ-4, ДМДВ-5.

ПО «ГМП-26» разработано для операционной среды (далее ОС) Linux. В ходе разработки проверялась работа в ОС: Linux-Mint (19.3 и 20), Astra Linux (Orel и Smolensk).

Для работы ПО необходима следующая конфигурация ЭВМ:

Процессор	x86-совместимый, 64-битный, с тактовой частотой не ниже 2 ГГц
ОЗУ	не менее 2 ГБ

Для работы ПО «ГМП-26» необходим набор финских или отечественных датчиков или программ имитирующих их работу. Состав датчиков следующий: 2 – датчика ветра (побортно), 2 – датчика температуры и влажности (побортно), 1 – датчик атмосферного давления, 1 – датчик нижней границы облачности, 1 – датчик метеорологической дальности видимости

2 Установка программного обеспечения

Дистрибутив программного обеспечения «ГМП-26» представляет собой архив GMP_26.rar В корневой директории после распаковки архива, формируется файлы:

config.ini, config.bin, gmp_config, gmp_linux, gmp_linux.ico. Проект написан на языке Free Pascal в среде Lazarus-ide. Версия Lazarus-ide, используемая для компиляции проекта -

2.0.10 (amd64). Для компиляции проекта необходимо установить библиотеку synapse (версии не ниже 40.1).

Основной задачей при разработке ПО «ГМП-26» является классифицирование задач, участвующих в работе программы, описанных в виде классов, сходство процессов отражено в виде отношения наследования и т.д. Некоторые алгоритмы реализованы в модулях-библиотеках (без образования классов). С описанием всех модулей проекта можно ознакомиться в приложении 1 («Структура проекта»).

Важной в алгоритмах программы является концепция актуального значения, она описана в приложении 2 («Концепция актуального значения»)

В общем виде программа работает в асинхронном режиме. Процессы получения данных, отображения и рассылки сообщений происходят независимо. Это обеспечивает стабильность временных характеристик (например программа не зависнет, если от датчика не поступит сообщение) и отзывчивость интерфейса программы.

Потенциальные источники исключений (редактирование телеграммы и печать протоколов испытаний) вынесены в отдельные процессы (отдельные исполняемые файлы) для повышения общей стабильности.

В программе предусмотрен механизм восстановления связи с устройствами (датчиками и адресатами сообщений) в случае разрыва соединения.

В программе предусмотрена защита от несанкционированной настройки. Для этого файл config.bin защищен шифрованием. Работать с файлом можно с помощью утилиты cryptor.

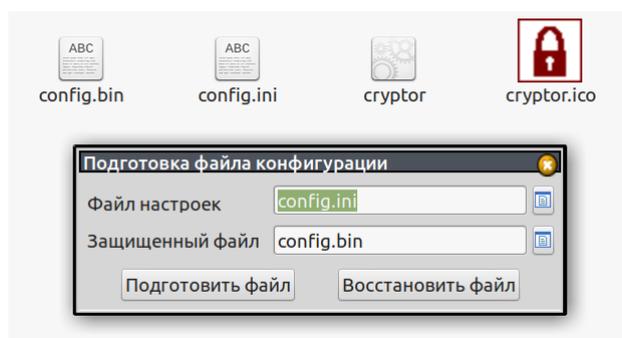


Рисунок 2 — интерфейс программы cryptor

Утилита позволяет подготовить защищенный файл на основе конфигурационного ini-файла, а так же восстановить ini-файл из защищенного. Утилита может быть скомпилирована для ОС Linux.

3 Запуск программного обеспечения

Для успешного запуска проекта нам необходимо будет создать ярлык на рабочем столе для этого

Программа «Файловый менеджер рабочего стола Fly» — штатное средство просмотра каталогов ОС Astra Linux не запускает программы по двойному щелчку мыши (как это обычно происходит в ОС Windows). Для запуска приложения необходимо создать ярлык на рабочем столе.

Для начала нужно убедиться, что у исполняемого файла установлены права на запуск (значок «ракета»). Выбираем файл, вызываем контекстное меню (правая кнопка мышки), выбираем пункт меню «Свойства».

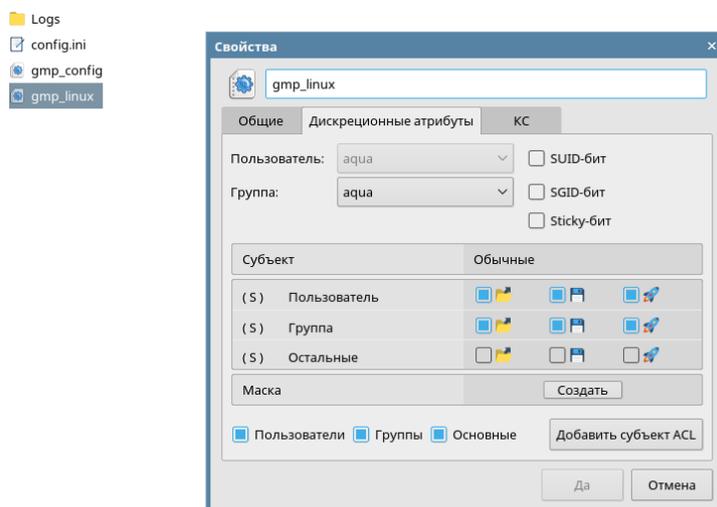


Рисунок 1 — Свойства файла

Выделяем файл и вызываем контекстное меню (правая кнопка мыши), выбираем пункт меню «Отправить», затем «Рабочий стол (создать ярлык)»

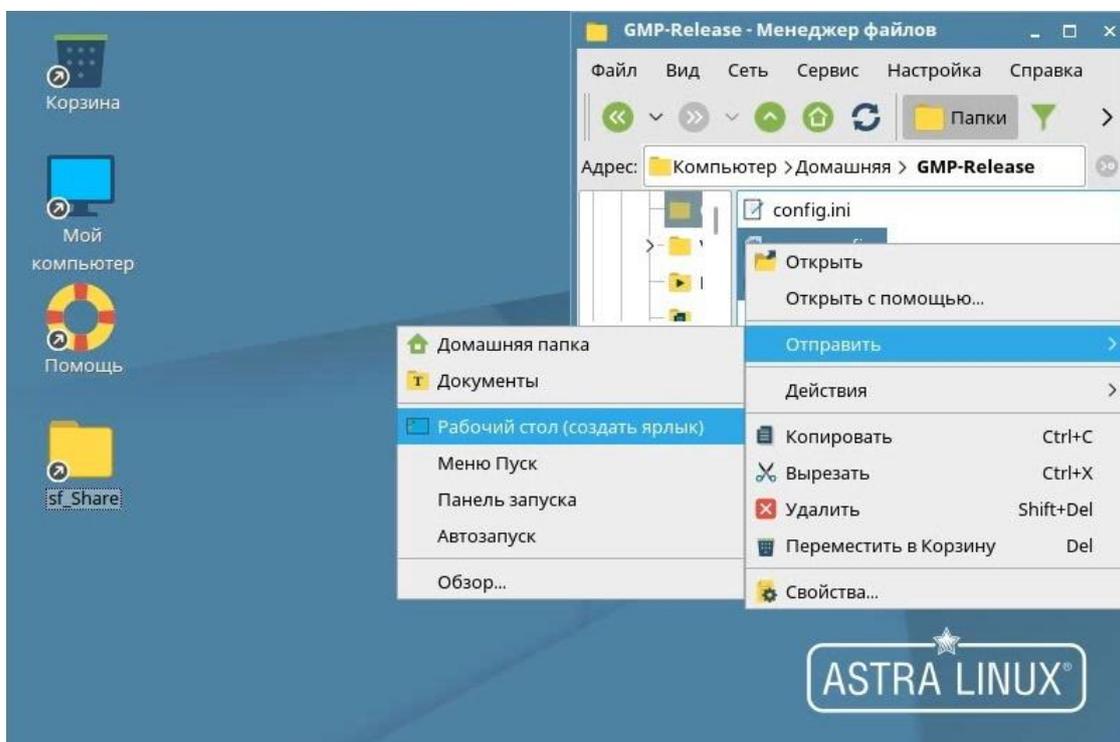


Рисунок 2 — Создать ярлык

Аналогично открываем свойства ярлыка на рабочем столе и переходим на вкладку «Ярлык», затем меняем тип на «Приложение»

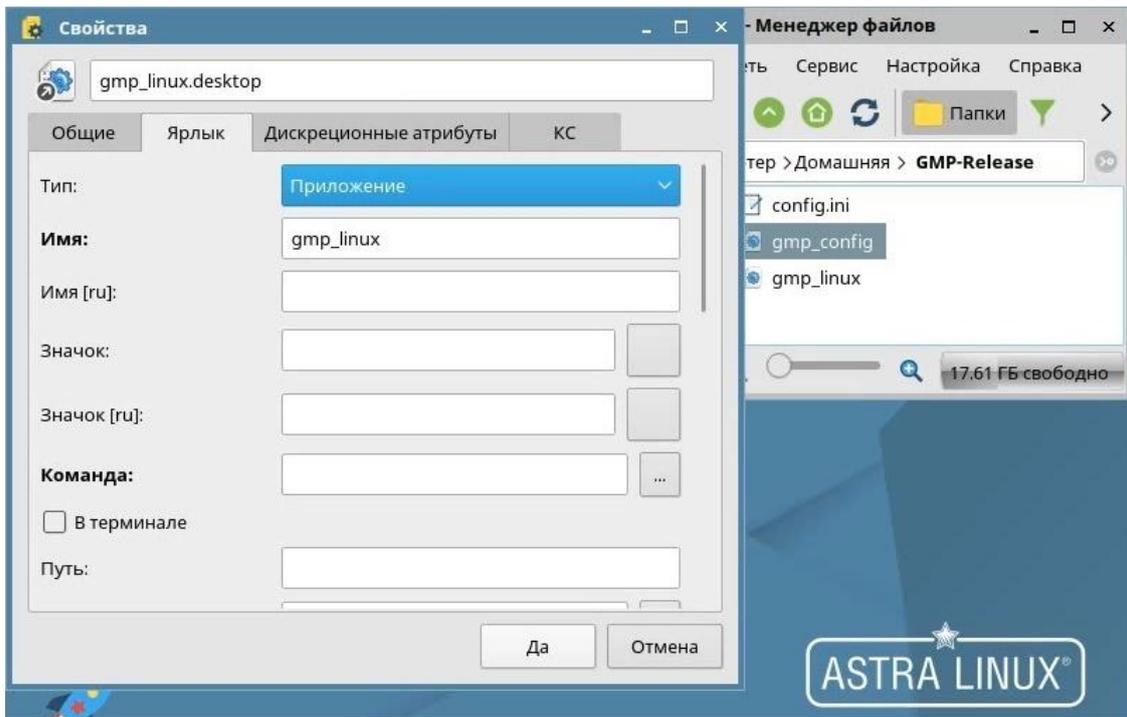


Рисунок 3 — Тип ярлыка

Размещаем окна так, чтобы одновременно видеть и окно свойств ярлыка и окно каталога с программой, как это показано на рисунке 4. Затем перетаскиваем исполняемый файл из его каталога в поле «Команда». Поле должно заполниться адресом исполняемого файла.

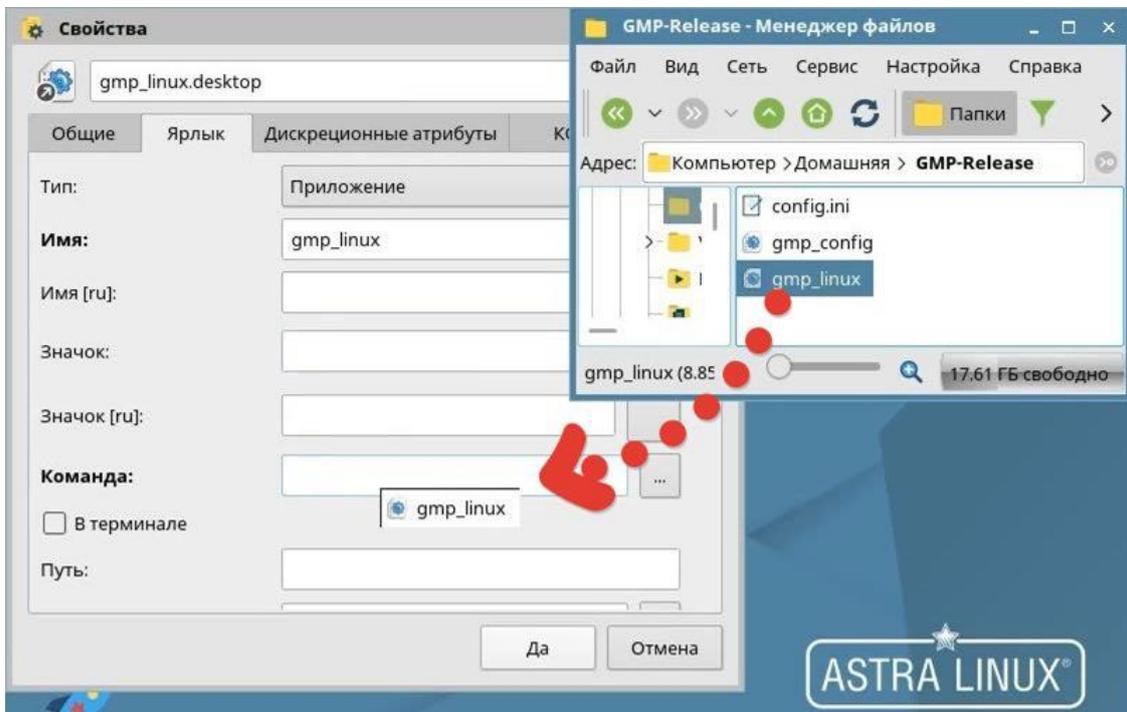


Рисунок 4 — Команда

Удаляем из поля «Команда» начальный текст `file://`, один «слеш» (`«/»`) должен остаться

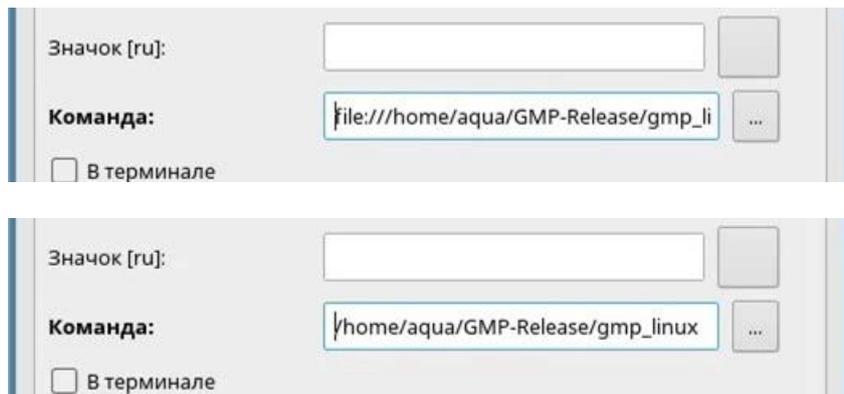


Рисунок 5 — Редактирование команды

Копируем часть команды от начала до последнего слеша, не включая его — это рабочий каталог программы. На рисунке 6 выделена строка /home/aqua/GMP-Release



Рисунок 6 — Копируем путь

Заполняем с помощью скопированного фрагмента поле «Путь»



Рисунок 7 — Заполняем путь

При желании можно указать файл gmp_linux.png как значок, тогда ярлык на рабочем столе сменит значок с ракеты на более подходящий прибор по измерению ветра

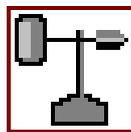


Рисунок 8 — Значок программы

После всех манипуляций необходимо нажать на кнопку «Да» внизу формы



Рисунок 9 — Завершение настроек

