

ООО „КРЮГЕР ХАУС“

Документация по блоку обмена между Artix и УНФ в части кассовых смен
Конфигурация УНФ 3.0



Новосибирск, 2023 г.

Оглавление

	Стр
1 Установка и подготовка к использованию	1
1.1 Начальные (текущие) системные требования	1
1.2 Установленные пакеты	1
1.3 Установка	1
1.4 Настройка запуска по расписанию	2
2 Алгоритм	4
3 Установка ПО	6

1 Установка и подготовка к использованию

1.1 Начальные (текущие) системные требования

- * Процессор - Intel(R) Atom(TM) CPU D2500 @ 1.86GHz, 2 ядра
- * ОЗУ - 2 Гб
- * SSD - 120 Гб
- * Операционная система (не ниже) - Ubuntu 22.04 jammy

1.2 Установленные пакеты

- * Python версии ≥ 3.11
- * Webmin (не обязательно) - панель для администрирования сервера
- * OpenSSH (установка 3.0.1.)
- * редактор Nano (установка 3.0.2.)

1.3 Установка

- Копируем каталог «Workshift_load» с программой на рабочий сервер
- Заходим в каталог «src» и выполняем команду

```
chmod +x wsh_load.py
```

Для того, что бы сделать файл скрипта исполняемым, в противном случае он не будет запускаться.

- Переходим в корневой каталог

```
cd ..
```

- В корневом каталоге выполняем команду для создания виртуального окружения

```
python3.11 -m venv .venv
```

- Выполняем команду для установки нужных пакетов

```
pip install -r requirements.txt
```

1.4 Настройка запуска по расписанию

- Выполняем команду для запуска планировщика

```
crontab -e
```

- В открывшемся редакторе в конец файла добавить строку

```
* * * * * /home/administrator/Workshift_load/src/wsh_load.py $HOME/command.log 2>&1
```

Здесь мы указываем периодичность запуска, полный путь к исполняемому скрипту, путь до файла в который будут выводиться сообщения планировщика. Сохранить файл. Теперь при таких настройках скрипт будет запускаться каждую минуту. Для работы нужно выставить нужный интервал запуска. Он выставляется в первой секции строки:

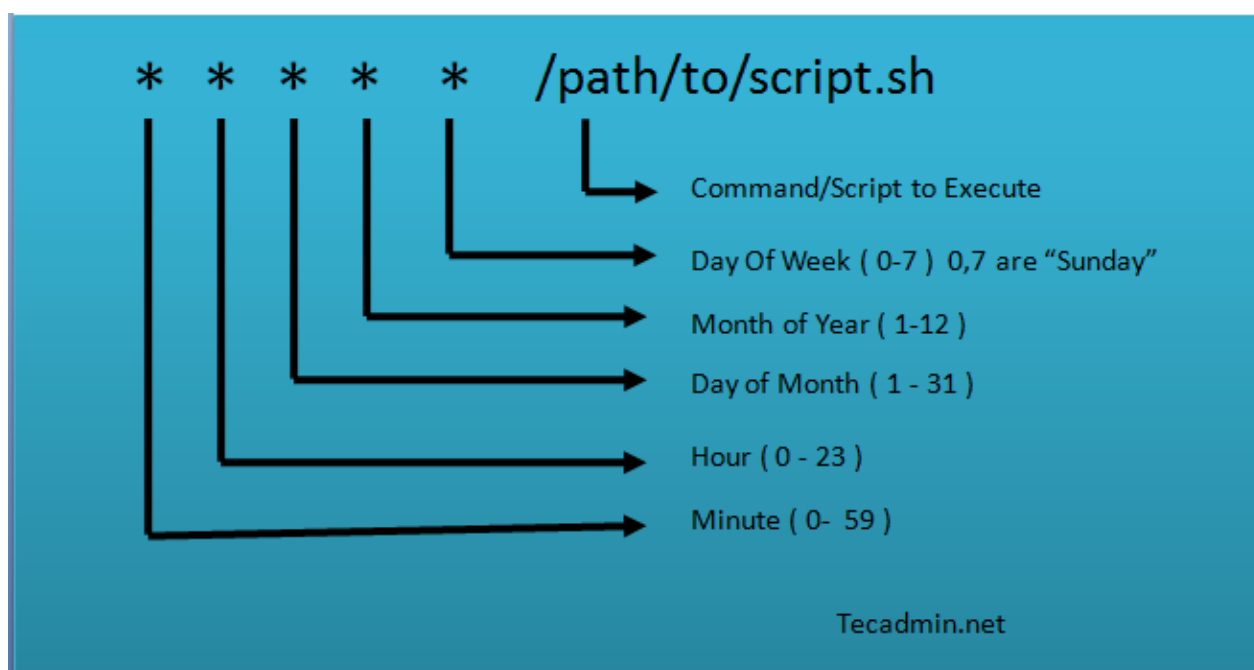


Рисунок 1.1 – «Формат Linux Crontab».

Синтаксис: `*(Minute)*(Hour)*(Day of the Month)*(Month of the Year) *(Day of the Week) username <path to command/script to execute>`

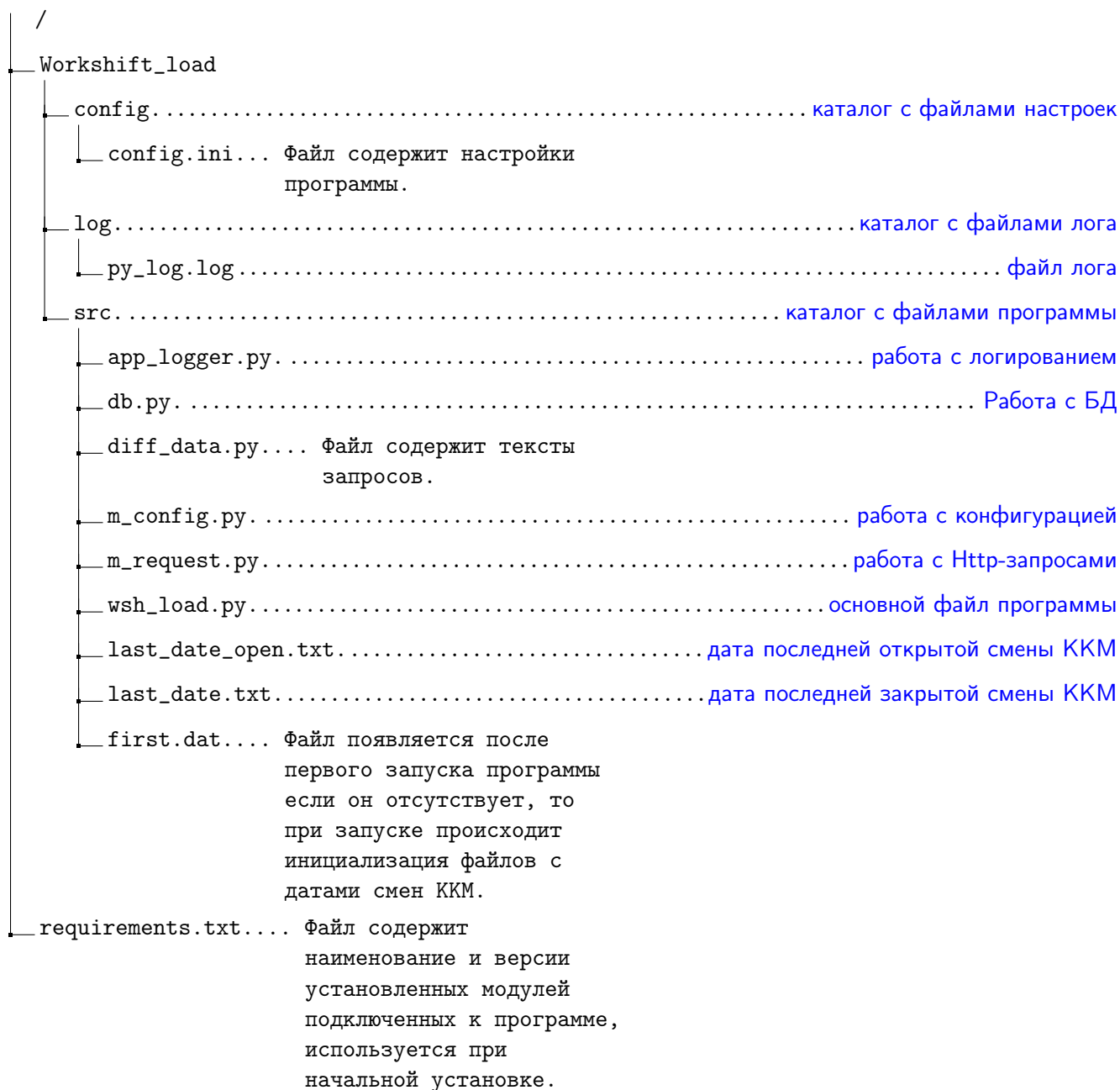
Минуты	Это значение может быть в пределах 0 — 59
Часы	Это значение может быть в пределах 0 — 23
День месяца	Это значение может быть в пределах 1 — 31
Месяц в году	Это значение поля находится в диапазоне от 1 до 12. Так же можно использовать три первые буквы названия месяца, например: jan, feb, mar
День недели	Это значение поля находится в диапазоне от 0 до 7. Где 0 и 7-воскресенье. 1-понедельник, 2-вторник и так далее

Пример:

Следующее выражение для выполнения задачи каждые 5 минут.

```
*/5 * * * * /home/administrator/Workshift_load/src/wsh_load.py
```

Структура проекта



2 Алгоритм

- При запуске программы происходит чтение настроек. Если чтение успешно, запускается функция «main()», в противном случае программа завершает свою работу.

```
if __name__ == "__main__":

    # Чтение настроек
    m_conf = m_config.m_Config()
    rc = m_conf.loadConfig()
    if not rc == None:
        main()
    else:
        logger.info(u'Программа завершила работу')
```

- При запуске функции «main()» происходит соединение с базой данных и создается объект для работы с ней, так же создается объект для работы с Http запросами.

```
tData = db.workDb(rc)
rec_con = m_request.req1C(rc)
```

- Получаем список смен которые были открыты с последней зафиксированной даты, если появились новые открытые смены, тогда формируем и отправляем Http запрос в 1С. Если код возврата был успешным (200), тогда меняем дату в файле, на дату открытия последней смены.

```
# Список открытых смен от последнего зафиксированного времени
l_workshift_open = tData.get_last_workshift_open()
# Если нечего отправлять, то не отправляем
if len(l_workshift_open) > 0:
    status_code = rec_con.post_workshift_open(l_workshift_open)
# Меняем дату в файле только в случае успешного результата работы 1С
if status_code == 200:
    tData.save_new_date_open()
else:
    logger.info(u'status_code_open - ' + str(status_code ))
```

- Получаем список смен которые были закрыты с последней зафиксированной даты, если появились новые закрытые смены, тогда формируем и отправляем Http запрос в 1С. Если код возврата был успешным (200), тогда меняем дату в файле, на дату закрытия последней смены.


```
# Список закрытых смен от последнего зафиксированного времени
l_workshift = tData.get_last_workshift()
# Если нечего отправлять, то и не отправляем
if len(l_workshift) > 0:
    status_code = rec_con.post_workshift(l_workshift)
# Меняем дату в файле только в случае успешного результата работы С1
    if status_code == 200:
        tData.save_new_date()
    else:
        logger.info(u'status_code - ' + str(status_code ))
```

- Завершаем работу программы.

3 Установка ПО

3.0.1. Установка OpenSSH

```
sudo apt install openssh-client openssh-server
```

3.0.2. Установка Nano

```
sudo apt install nano
```

Список иллюстраций

1.1	«Формат Linux Crontab».	2
-----	-------------------------	---

Журнал изменений

Версия	Дата	Автор(ы)	Изменения
0.01	27.01.23	РК	Начало работы
0.1	31.01.23	РК	Корректировка