

## **Описание и настройка программного обеспечения однонаправленной передачи файлов SFT\_multi. Версия описания 4.2**

Программное обеспечение (ПО) SFT\_multi – это комплект утилит, предназначенных для передачи файлов через однонаправленный шлюз. ПО SFT\_multi позволяет просматривать содержимое любого количества «корневых» папок на компьютере внешней сети и производить однонаправленную передачу файлов из них во внутреннюю сеть.

В качестве однонаправленного шлюза рассматривается одно из следующих устройств:

- однонаправленный шлюз 3-Кросс;
- однонаправленный шлюз СТРОМ-100;
- высокоскоростной однонаправленный шлюз СТРОМ-1000.

ПО однонаправленной передачи SFT\_multi платфрмо-независимое. Все утилиты под любую операционную систему (ОС) совместимы между собой. Это позволяет использовать на компьютерах в соединяемых сетях как одинаковые, так и разные операционные системы - это никак не влияет на процесс передачи файловой информации.

## Основные требования.

Для однонаправленной передачи файлов помимо однонаправленного шлюза необходимо использовать два персональных компьютера (ПК), расположенных в соединяемых сетях, которые в дальнейшем будем называть внешний ПК (PC\_ext) и внутренний ПК (PC\_int). Пример типовой структуры включения приведён на Рис.1.



Рис.1. Структура сети для передачи файлов.

Основные требования к оборудованию:

- одинаковые по производительности ПК во внешней и внутренней сети;
- два сетевых интерфейса в каждом ПК;
- для внутреннего ПК: наличие сетевой карты с оптическим интерфейсом, либо наличие оптического конвертера со следующими параметрами:
  - для шлюзов 3-Кросс и СТРОМ-100 – оптическая технология физического уровня 100BASE-FX (пропускная способность до 100 Мбит/с, многомодовый оптоволоконный кабель, длина волны 1310 нм);
  - для шлюза СТРОМ-1000 – оптическая технология физического уровня 1000BASE-SX (пропускная способность до 1 Гбит/с, многомодовый оптоволоконный кабель, длина волны 850 нм);
- Функционирование под управлением любой ОС из следующего списка:
  - Linux Debian 32 bit;
  - Linux Debian 64 bit;
  - Windows 32 bit;
  - Windows 64 bit;
  - Заря 1.1;
  - Astra Linux (ver 1.4, ver 1.5);
  - MCBC 5.0;
  - MCBC 3.0 (ВИ 12, ВИ 13, ВИ 14, ВИ 16).

## Структура и функционирование программного обеспечения.

В состав ПО однонаправленной передачи файлов входят две утилиты:

**ScanSendService** — утилита для передачи файлов. Устанавливается на внешнем ПК. При запуске ScanSendService следит за появлением файлов в «корневых» папках внешнего ПК, указанных в конфигурационном файле. В случае появления в одной из них файла утилита его блокирует и обрамляет необходимыми заголовками. Далее она режет данные на UDP-пакеты и отправляет их на указанный в файле конфигурации порт и IP-адрес. После этого файл из «корневой» папки внешнего ПК удаляется.

**RecvMultiPort** - утилита для приема файлов. Устанавливается на внутреннем ПК. При запуске утилита постоянно «слушает» указанные в конфигурации порты. В случае приема заголовка требуемой структуры RecvMultiPort пытается собрать из поступивших данных файл. Если контрольные суммы принятого и отправленного файла совпадают, то файл записывается в указанную в файле конфигурации «корневую» папку на внутреннем ПК. Если файл собрать не удалось, то в «корневую» папку ничего не записывается, в логе отображается запись о неудачном приеме.

### Замечания по работе программного обеспечения.

1. ScanSendService не передает пустые папки.
2. При запуске утилит системой и их работе в фоновом режиме без непосредственного взаимодействия с пользователем (программа демон, daemon (dæmon) mode, бесконсольный режим) поиск файла конфигурации происходит в следующем порядке:
  - рабочая папка;
  - папка с запускаемым файлом;
  - /etc (в Linux системах).
3. Для исключения сбоев передачи рекомендуется устанавливать статическую ARP-запись на IP-адрес компьютера назначения, куда будут отправляться файлы.
4. Пример консольного запуска утилиты с параметрами:

```
./ScanSendService -m:1 -t:1400 -s:95
```

## Настройка и параметры программного обеспечения.

### Настройка ScanSendService.

После первого запуска утилиты ScanSendService без параметров в рабочей папке создается конфигурационный файл scansend\_cfg и файл списка сканирования scan\_folder.lst. При каждом следующем запуске программа будет считывать параметры из файлов scansend\_cfg и scan\_folder.lst. Если ScanSendService запускается с ключом, который так же указан в файле scansend\_cfg, то ключ в файле конфигурации scansend\_cfg игнорируется.

### Параметры конфигурационного файла scansend\_cfg.

m: Режим запуска.

При m:0 утилита просканирует ОДИН раз все «корневые» папки, отправит находящиеся там файлы и закончит свою работу.

При m:1 утилита будет постоянно проверять все «корневые» папки на наличие файлов и осуществлять их передачу.

При m:2 утилита будет постоянно проверять все «корневые» папки на наличие файлов и осуществлять их передачу. При этом работа будет происходить в бесконсольном режиме (daemon mode).

При `m:3` производится остановка бесконсольного режима (`daemon stop`).

`v`: Вывод на консоль.

При `v:0` ничего не выводится.

При `v:1` осуществляется вывод на консоль.

При `v:2` осуществляется расширенный вывод на консоль (режим отладки).

`l`: Ведение лога.

При `l:1` логирование включено.

При `l:0` логирование выключено.

`g`: Задаёт абсолютный путь к лог-файлу.

Если параметр не задан, то лог-файл будет находиться в папке запуска утилиты.

Обязательно надо указывать путь к лог-файлу, если утилита запускается в `daemon mode`.

`f`: Задаёт «корневую» папку для отслеживания файлов.

Параметр работает, если не указан файл списка сканирования.

Пример использования: `f:scan_folder`

`i`: Задаёт IP-адрес, на который будет отправляться файл.

Параметр работает, если не указан файл списка сканирования.

Пример использования: `i:192.168.1.5`.

`p`: Задаёт порт назначения.

Параметр работает, если не указан файл списка сканирования, по умолчанию задан 55001.

`u`: Задаёт количество повторов заголовка.

Для повышения надёжности на случай потери пакетов при передаче через шлюз. При получении этой информации утилита на принимающей стороне пытается собрать файл.

По умолчанию параметр равен 5.

`d`: Задаёт количество передач данных.

Для повышения надёжности передачи данных реализовано дублирование этой операции.

При `d:1` данные передаются один раз.

При `d:2` два раза.

При нормальной работе достаточно `d:1`. По умолчанию параметр равен 2.

`s`: Задаётся скорость передачи в Мбит/с.

Для пропускной способности 100 Мбит/с рекомендованное максимальное значение 95; для пропускной способности 1 Гбит/с максимальное значение то, которое способно принять/передать используемое оборудование, максимальное рекомендованное значение - 960.

`t`: Минимальный размер посылки в байтах.

Возможны значения от 1400 до 65507. При `t:1400` данные режутся по 1 пакету UDP, фрагментированные пакеты отсутствуют, достижение максимальной скорости передачи маловероятно. При больших значениях параметра будут фрагментированные пакеты, и будет достигаться максимальная скорость передачи.

Рекомендованные значения `t:1400` или `t:61440`.

Выбор значения 61440 объясняется тем, что в этом случае уменьшается нагрузка на дисковую подсистему. Общая рекомендация выбирать этот ключ кратным 4 Кбайтам.  $61400 = 4\text{Кбайт} * 60$  – максимально возможное значение параметра в указанном диапазоне,

кратное 4 Кбайтам.

**j**: Задает путь к файлу списка сканирования.

Пример использования: `j:/home/unidir/scan_folder.lst`

**c**: Время ожидания перед следующим сканированием папок в секундах.

**b**: Исключает из передачи файлы, в названии которых присутствует указанный префикс.

Например, `b:.` исключает из передачи файлы, название которых начинается с точки.

**e**: Исключает из передачи файлы с указанным расширением.

**z**: Исключает из передачи файлы нулевой длины.

При `z:1` опция включена.

**x**: Задает задержку в секундах отправки файла от момента его записи в исходную папку.

Опция используется для контроля того, что файл в папку источник записался полностью, и мы не начнем его отправку раньше времени окончания записи. ПО таким образом следит за приходящими файлами. Основано на особенностях функционирования служб FTP, связанных с установкой времени создания файла. По умолчанию значение равно 0.

**a**: Предустанавливает значение локали (язык, кодировка, страна) (используется только в Linux системах).

Когда утилита запускается в фоновом режиме (daemon mode) возможна ситуация - служба ПО стартовала, но значение локали не доступно. Это приводит к тому, что передача файлов, в названии которых содержатся русские буквы, может быть невозможна. Предустановка локали решает эту проблему. Рекомендованное значение `a:UTF-8`

**k**: Включение работы «корзины» для переданных файлов.

При `k:1` включено.

При `k:0` выключено, по умолчанию задается эта величина параметра.

**y**: Задает путь к «корзине».

Работает при `k:1`.

**o**: Задает имя конфигурационного файла.

По умолчанию используется `scansend_cfg`

**h**: Помощь.

Краткое описание ключей.

### **Структура файла списка сканирования.**

Файл представляет собой строки следующей структуры:

`<IP_dst>:<Port_dst> <path_to_scan_folder> <user_name>`

Параметр `<IP_dst>` задает IP-адрес, на который будет отправляться файл, поле `<Port_dst>` используется для задания порта назначения, параметр `<path_to_scan_folder>` задает «корневую» папку для отслеживания файлов, поле `<user_name>` указывать необязательно, если оно задано, на приемной стороне в папке, к которой «привязан» `<Port_dst>`, будет создана вложенная папка с именем `<user_name>`

Пример файла списка сканирования:

`192.168.10.1:55001 /home/usr1 usr1`

`192.168.10.1:55001 /home/usr2 usr2`

```
192.168.10.1:55003  usr3
192.168.10.1:55004  usr4
192.168.10.1:55008  usr5
```

### **Настройка RecvMultiPort.**

После первого запуска утилиты RecvMultiPort без параметров в рабочей папке создается конфигурационный файл `recvmultiports_cfg`, файл сканируемых портов `scan_ports.lst` и папка для временных файлов `tmp`. При каждом следующем запуске RecvMultiPort будет считывать параметры из `recvmultiports_cfg` и `scan_ports.lst`. Если утилита запускается с ключом, который так же указан в файле `recvmultiports_cfg`, то ключ в файле конфигурации `recvmultiports_cfg` игнорируется.

### **Параметры конфигурационного файла `recvmultiports_cfg`.**

**m:** Режим запуска.

При **m:1** запускается консольный режим работы.

При **m:2** работа утилиты происходит в фоновом режиме (бесконсольный режим, `daemon mode`).

При **m:3** осуществляется остановка бесконсольного режима (`daemon stop`)

**v:** Вывод на консоль.

При **v:0** ничего не выводится.

При **v:1** осуществляется вывод на консоль.

При **v:2** осуществляется расширенный вывод на консоль (режим отладки).

**l:** Ведение лога.

При **l:1** логирование включено.

При **l:0** логирование выключено.

**g:** Задаёт абсолютный путь к лог-файлу.

Если параметр не задан, то лог-файл будет находиться в папке запуска утилиты. Обязательно надо указывать путь к лог-файлу, если утилита запускается в `daemon mode`.

**t:** Задаёт путь к папке для временных файлов.

**p:** Задаёт сканируемый порт.

Параметр работает, если не указан файл сканируемых портов, по умолчанию используется значение 55001.

**d:** Задаёт название папки для принятых файлов.

Параметр работает, если не указан файл сканируемых портов.

**c:** Задаёт таймаут неудачи по получению файла (в мс).

По умолчанию 15000.

**r:** Задаёт таймаут удаления из очереди полученного файла (в мс).

По умолчанию 5000.

**j:** Задаёт путь к файлу сканирования портов.

**n:** Включение/Выключение режима отладки для оценки производительности системы.

При **n:1** включается режим отладки. В этом режиме происходит приём файлов, но не производится их сохранение на диск. Таким образом ПО работает без затрат ресурсов на

дисковую подсистему. Данный режим позволяет оценить производительность системы по приёму данных из сети, исключая дисковую подсистему.

**s** : Задаёт режим записи полученной информации на диск.

При **s : 1** выполнение команды `sync` (запись информации из кэша на диск) для дисковой подсистемы производится при получении каждого блока данных.

При **s : 0** команда `sync` для дисковой подсистемы производится только при успешном приёме всего файла целиком.

Данную опцию рекомендуется включать при использовании систем с невысокой производительностью. Значение по умолчанию **s : 0**.

**a** : Предуславливает значение локали (язык, кодировка, страна) (используется только для Linux систем).

Когда утилита запускается в фоновом режиме (`daemon mode`) возможна ситуация - служба ПО стартовала, но значение локали не доступно. Это приводит к тому, что передача файлов, в названии которых содержатся русские буквы, становится невозможной. Для решения этой проблемы рекомендуется задавать значение параметра **a : UTF-8**

**k** : Включение работы «корзины» для переданных файлов.

При **k : 1** включено.

При **k : 0** выключено, по умолчанию задается этот параметр.

**y** : Задаёт путь к «корзине».

Работает при **k : 1**.

**x** : Устанавливает права для сохраняемых файлов (используется только для Linux систем).

По умолчанию `777`.

**o** : Задаёт имя конфигурационного файла.

По умолчанию используется `recvmultiports_cfg`

**h** : Помощь.

Краткое описание ключей.

### **Структура файла сканирования портов.**

Файл представляет собой строки следующей структуры:

`<Port_dst> <path_to_destination_folder>`

Поле `<Port_dst>` используется для задания сканируемого порта, параметр `<path_to_destination_folder>` задаёт папку назначения для принятых файлов, которая ассоциируется с данным портом.

Пример файла сканирования портов:

`55001 dest`

`55002 dest2`

## Пример настройки программного обеспечения.

**Задача.** Необходимо реализовать однонаправленную передачу файлов между двумя сегментами сети. Во внешней сети выделен IP-адрес 10.8.0.5. Во внутренней сети выделен IP-адрес 10.8.1.2. На ПК с выделенными IP-адресами должны быть организованы папки для общего доступа user1, user2, user3 (Рис.2.).

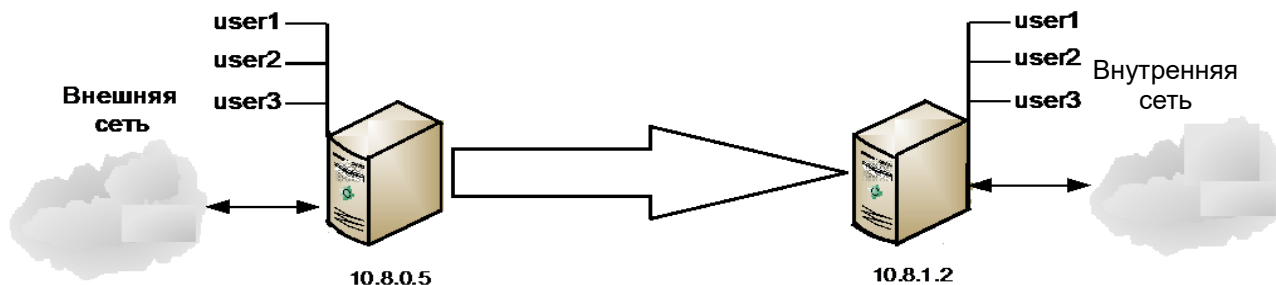


Рис.2. Исходные данные.

**Реализация.**

### 1. Выбор аппаратных компонент.

Для решения данной задачи необходимо следующее оборудование (Рис.3.):

- 1.1 шлюз однонаправленной передачи данных.
- 1.2 ПК одинаковой производительности PC\_ext и PC\_int.
- 1.3 у ПК PC\_ext должно быть 2 сетевых интерфейса. Один из них - витая пара для подключения к однонаправленному шлюзу (на Рис.3. это eth0), второй – для подключения к внешней сети (на Рис.3. это eth1).
- 1.4 у ПК PC\_int должно быть 2 сетевых интерфейса. Один из них - с оптическим интерфейсом (на Рис.3 это eth0). Либо конфигурацию необходимо дополнить оптическим конвертером.
- 1.5 ОС на ПК — Linux Debian.

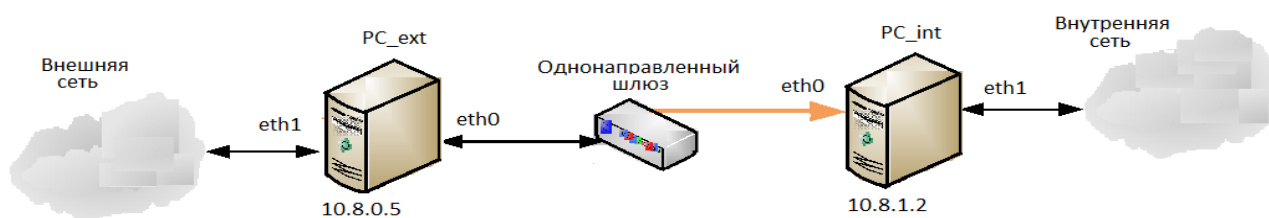


Рис.3. Аппаратные компоненты.

### 2. Распределение сетевых адресов.

Для интерфейсов, которые соединяются с однонаправленным шлюзом выбираются адреса, не пересекающиеся с адресами сетевых интерфейсов, к которым подключается сеть (Рис.4).

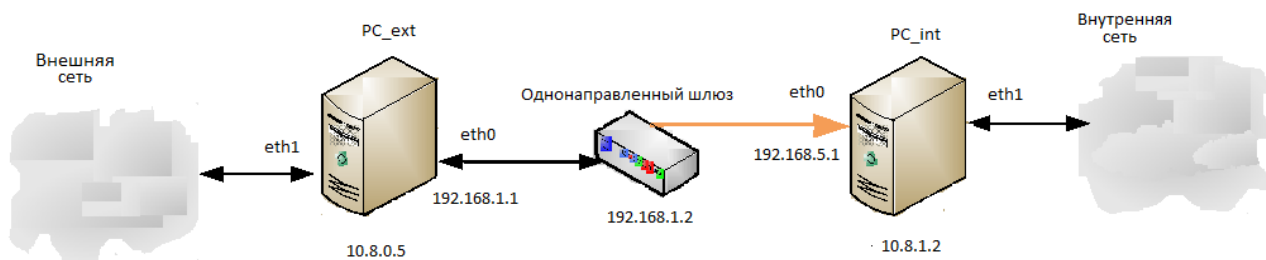


Рис.4. Распределение IP-адресов.



### 3. Распределение MAC-адресов.

3.1 назначаем MAC-адрес у однонаправленного шлюза.

MAC-адрес однонаправленного шлюза: 00:11:22:33:44:55 (назначается)

3.2 определяем MAC-адрес у интерфейса eth0 ПК PC\_int. Это осуществляется командой `ifconfig eth0`.

Допустим, что MAC-адрес сетевого интерфейса eth0 : 00:23:F2:AB:23:33.

### 4. Создаем файл конфигурации для однонаправленного шлюза.

Исходя из полученной структуры сети, создаем конфигурационный файл (`z.txt`).

4.1 указываем MAC-адрес однонаправленного шлюза

```
wan mac 00:11:22:33:44:55
```

4.2 указываем разрешенный IP-адрес

```
ip permit 192.168.1.1
```

4.3 указываем маршрутизацию в закрытую сеть

```
route 192.168.1.2 192.168.5.1 00:23:F2:AB:23:33
```

4.4 компилируем конфигурационный файл

```
./zc z.txt z.cfg
```

4.5 полученный файл `z.cfg` сохраняем на SD-карту и загружаем в однонаправленный шлюз.

Подробности можно найти в «Описание конфигурирования однонаправленного шлюза .....pdf»

### 5. Проверка взаимодействия с однонаправленным шлюзом.

5.1 на PC\_ext запустить пинг на 192.168.1.2:

```
ping 192.168.1.2
```

при этом ответов не будет (это нормально)

5.2 проверить, появилась ли ARP-запись на адресе 192.168.1.2:

```
arp -a
192.168.1.2  -----  00:11:22:33:44:55
```

адресу 192.168.1.2 должен соответствовать MAC-адрес однонаправленного шлюза

5.3 так же должен увеличиваться счетчик на дисплее однонаправленного шлюза.

5.4 помимо этого должен увеличиваться счетчик принятых пакетов у PC\_int на интерфейсе eth0 (команда `ifconfig eth0`, поле `RX packet` в ответе на команду).

Если пункты 5.2-5.4 выполняются, то взаимодействие с однонаправленным шлюзом налажено.

Иначе следует обратиться к «Описание конфигурирования однонаправленного шлюза .....pdf», раздел «Диагностика работоспособности».

### 6. Настройка ScanSendService на PC\_ext

**Настройка производится с правами root.**

6.1 создать папки:

```
/home/unidirect/user1  
/home/unidirect/user2  
/home/unidirect/user3
```

6.2 создать общий доступ для папок user1, user2, user3.

6.3 скопировать в /usr/sbin/ утилиту ScanSendService.

6.4 раздать права:

```
chmod +x /usr/sbin/ScanSendService
```

6.5 запустить ScanSendService.

6.6 появившийся файл конфигурации scansend\_cfg отредактировать следующим образом (не забываем добавить ключ g):

```
# Режим запуска:  
m:1  
# Дополнительная индикация:  
v:2  
# Ведение лога:  
l:1  
# Количество повторов заголовка:  
u:5  
# Количество повторов для данных:  
d:1  
# Скорость передачи MBit (0 - max) для Strom-1000:  
s:900  
# Размер MTU (min 1400..max 65507):  
t:61440  
# Файл списка сканирования:  
j:/etc/sft/scan_folder.lst  
# Задержка в секундах перед следующим сканированием:  
c:1  
x:0  
a:UTF-8  
g:/var/log/sft/sender.log
```

6.7 отредактировать scan\_folder.lst следующим образом:

```
192.168.1.2:55002 /home/unidirect/user1  
192.168.1.2:55003 /home/unidirect/user2  
192.168.1.2:55004 /home/unidirect/user3
```

6.8 переместить scan\_folder.lst в /etc/sft/

6.9 переместить scansend\_cfg\_cfg в /etc

## 7. Проверка работоспособности ScanSendService на PC\_ext

7.1 после корректировки конфигурации утилиты запустить ScanSendService:

```
ScanSendService
```

Утилита запустится в консольном режиме.

7.2 в папку /home/unidirect/user1 скопировать один или несколько файлов.

7.3 в случае нормальной работы ПО произойдет следующее:

7.3.1 скопированные файлы в /home/unidirect/user1 исчезнут;

7.3.2 увеличится счетчик на однонаправленном шлюзе;

7.3.3 увеличится счетчик принятых пакетов у PC\_int на интерфейсе eth0 (команда ifconfig eth0, поле RX packet в ответе на команду)

7.3.4 результат работы утилиты отобразится в консоли.

7.4 если пункты 7.3.1-7.3.4 выполняются, то ПО работает правильно.

7.5 если пункты 7.3.1-7.3.4 не выполняются, то необходимо искать проблему, рассматривая логи утилиты и ошибки, которые утилита пишет в эти логи.

## 8. Прописывание ScanSendService на PC\_ext в автозагрузку.

8.1. в файле /etc/scansend\_cfg изменяем режим запуска:

```
m:2
```

8.2. переименовываем файл стартового скрипта TX\_unidirect в unidirect и копируем файл в папку /etc/init.d

Файл TX\_unidirect находится в архиве с утилитами.

8.3. раздаем права:

```
chmod +x /etc/init.d/unidirect
```

8.4. прописываем скрипт в автозапуск системы:

```
update-rc.d unidirect defaults
```

8.5. перегружаем систему. После перезагрузки убеждаемся, что daemon-утилиты запущен и «висит» в процессах:

```
ps -A | grep Scan
2574 ..... ScanSendService
```

а также факт старта отображен в лог- файле /var/log/sft/sender.log:

```
Start ScanSendService Server.....
```

8.6. останавливать/стартовать/перезапускать фоновый режим работы утилиты можно средствами скрипта /etc/init.d/unidirect :

```
/etc/init.d/unidirect stop
/etc/init.d/unidirect start
/etc/init.d/unidirect restart
```

## 9. Настройка RecvMultiPort на PC\_int

### Настройка производится с правами root.

9.1 создать папки:

```
/home/user_int/user1
/home/user_int/user2
```

```
/home/user_int/user3
```

9.2 создать общий доступ для папок user1, user2, user3.

9.3 скопировать в /usr/sbin/ утилиту RecvMultiPort

9.4 раздать права:

```
chmod +x /usr/sbin/RecvMultiPort
```

9.5 запустить RecvMultiPort.

9.6 появившийся файл конфигурации recvmultiports\_cfg отредактировать следующим образом (не забываем добавить ключ g):

```
# Режим запуска:
m:1
# Дополнительная индикация:
v:2
# Ведение лога:
l:1
# Адрес папки для временных файлов:
t:/var/sft/tmp
# Таймаут неудачи по получению файла:
c:15000
# Таймаут удаления из очереди полученного файла:
r:5000
# Список сканируемых портов и их папок назначения:
j:/etc/sft/scan_ports.lst
a:UTF-8
g:/var/log/sft/reciver.log
```

9.7 отредактировать scan\_ports.lst следующим образом:

```
55002 /home/user_int/user1
```

```
55003 /home/user_int/user2
```

```
55004 /home/user_int/user3
```

9.8 переместить scan\_folder.lst в /etc/sft

9.9 переместить recvmultiports\_cfg в /etc

## 10. Проверка работоспособности RecvMultiPort на PC\_int

10.1 после корректировки файла конфигурации запустить RecvMultiPort:

```
RecvMultiPort
```

Утилита запустится в консольном режиме.

10.2 запустить утилиту ScanSendService на PC\_ext

10.3 скопировать на PC\_ext файл в папку user1

10.4 утилита RecvMultiPort должна среагировать на прием файла.

10.5 в случае удачной передачи — файл окажется в папке /home/user\_int/user1

10.6 далее этот файл можно забрать с использованием служб SMB-протокола пользователю из закрытой сети.

## 11. Прописывание RecvMultiPort на PC\_int в автозагрузку.

11.1 в файле `/etc/recvmultiports_cfg` изменяем режим запуска:

```
m:2
```

11.2 переименовываем файл стартового скрипта `RX_unidirect` в `unidirect` и копируем файл в папку `/etc/init.d` .

Файл `RX_unidirect` находится в архиве с утилитами.

11.3 раздаем права:

```
chmod +x /etc/init.d/unidirect
```

11.4 прописываем скрипт в автозапуск системы:

```
update-rc.d unidirect defaults
```

11.5 перегружаем систему. После перезагрузки убеждаемся, что демон утилиты запущен и «висит» в процессах:

```
ps -A | grep Recv
```

```
2574 ..... RecvMultiPort
```

а также факт старта отображен в лог-файле `/var/log/sft/reciver.log`:

```
Start RecvMultiPort Server .....
```

11.6 останавливать/стартовать/перезапускать фоновый режим работы утилиты можно средствами скрипта `/etc/init.d/unidirect` :

```
/etc/init.d/unidirect stop
```

```
/etc/init.d/unidirect start
```

```
/etc/init.d/unidirect restart
```