

Межканальная передача переменных в Asterisk

Обо мне:



- Компания **IQTEK**
- Разработка и внедрение на базе OSS VoIP
- Начинал с Asterisk 0.99
- Asterisk DCAP/DCAA
- Чёрный пояс по диалплану

О чем доклад

- Переменные Asterisk
- Способы и средства их передачи
- Доклад не претендующий на полноту
- Доклад претендующий на понимание

О чём не будем говорить

~~ODBC~~

~~MySQL~~

~~CURL~~

~~SHELL~~

~~AGI~~

Case 1: Dial(Local/...)

- Вызов оператора в Queue
- Необходимо задать дополнительную логику
 - Имя очереди
 - Дополнительные услуги

Case 1: Dial(Local/...)

- Вызов оператора в Queue
- Необходимо задать дополнительную логику
 - Имя очереди
 - Дополнительные услуги

```
same => n,Set(_QUEUE=100)  
same => n,Dial(Local/200@internal)
```

```
[internal]  
exten => 1,n,NoOp(_QUEUE)
```

Ответ: __ или _

- Переменные копируются
 - В момент создания
- _ - обрезаются, но __ - нет
- Перетирают друг друга

channel.c: void ast_channel_inherit_variables(...)

```
switch (vartype) {
case 1:
    newvar = ast_var_assign(&varname[1], ast_var_value(current));
    break;
case 2:
    newvar = ast_var_assign(varname, ast_var_value(current));
    break;
default:
    continue;
}
if (newvar) {
    ast_debug(1, "Inheriting variable %s from %s to %s.\n",
              ast_var_full_name(newvar), ast_channel_name(parent),
              ast_channel_name(child));
    AST_LIST_INSERT_TAIL(ast_channel_varshd(child), newvar, entries);
    ast_channel_publish_varset(child, ast_var_full_name(newvar),
                                ast_var_value(newvar));
}
```

Case 2: вернуть переменную обратно

- Поле вызова в Case1 зафиксировали результат
- Необходимо вернуть в CDR данные:
 - DIALSTATUS
 - Время вызова

Ответ: MASTER_CHANNEL()

- Доступна на чтение и запись
- Позволяет вызывать функции
- Обращается к старейшему каналу по **linkedid**

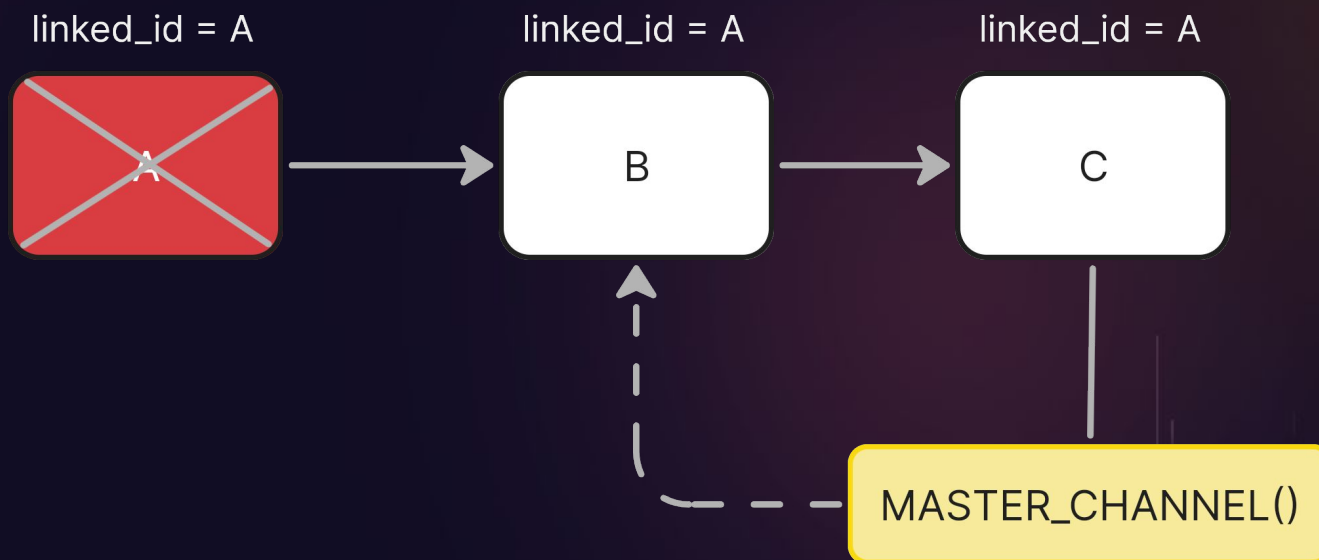
```
same => n,Set(MASTER_CHANNEL(BILLSEC)=100)
```

```
same => n,Set(MASTER_CHANNEL(CDR(accountcode))=client)
```

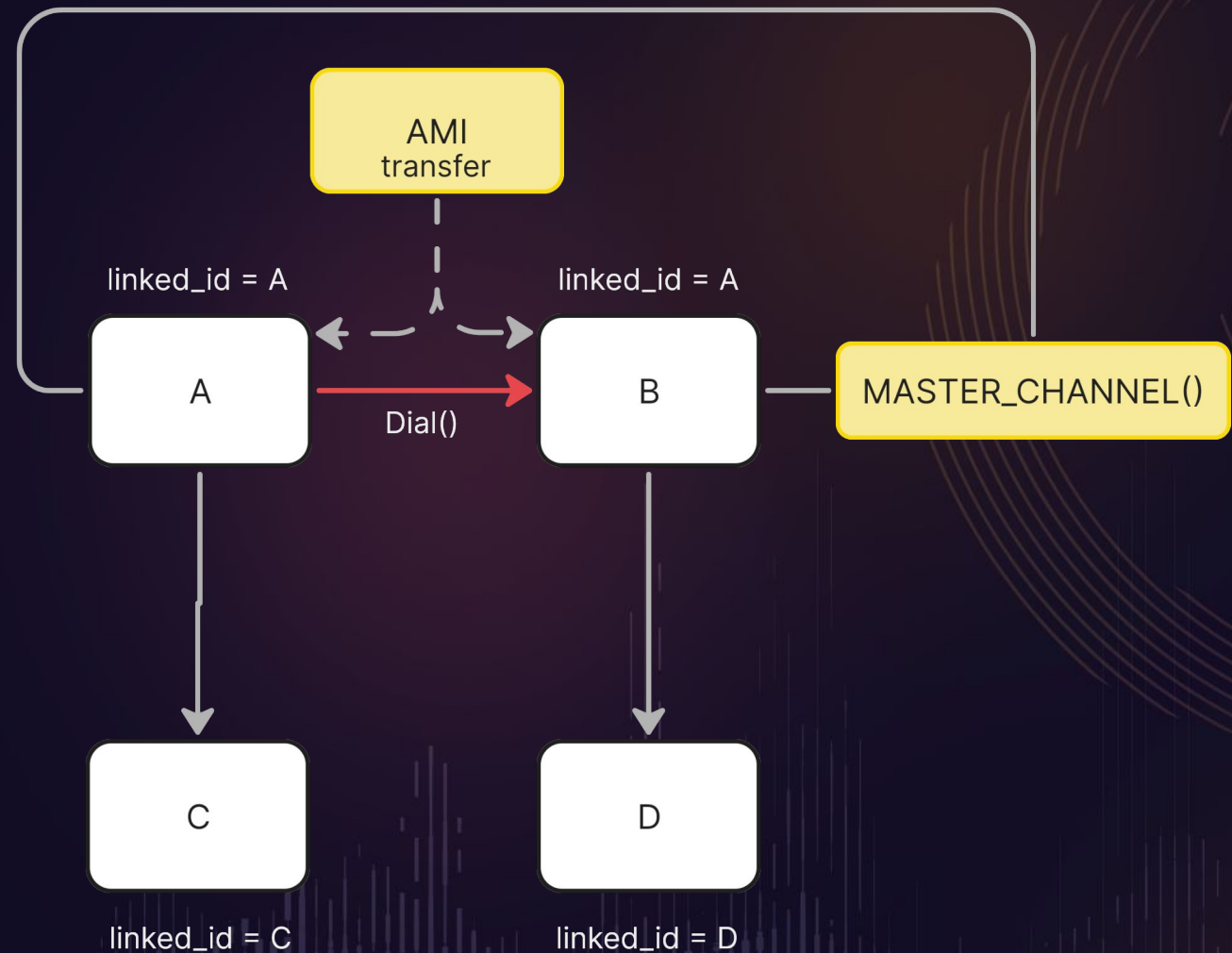
```
same => n,NoOP(${MASTER_CHANNEL(CHANNEL(name))})
```

```
same => n,NoOP(${MASTER_CHANNEL(_FOO)})
```

Отвѣт: MASTER_CHANNEL()



Отвѣт: MASTER_CHANNEL()



Ответ: MASTER_CHANNEL()

Следствия:

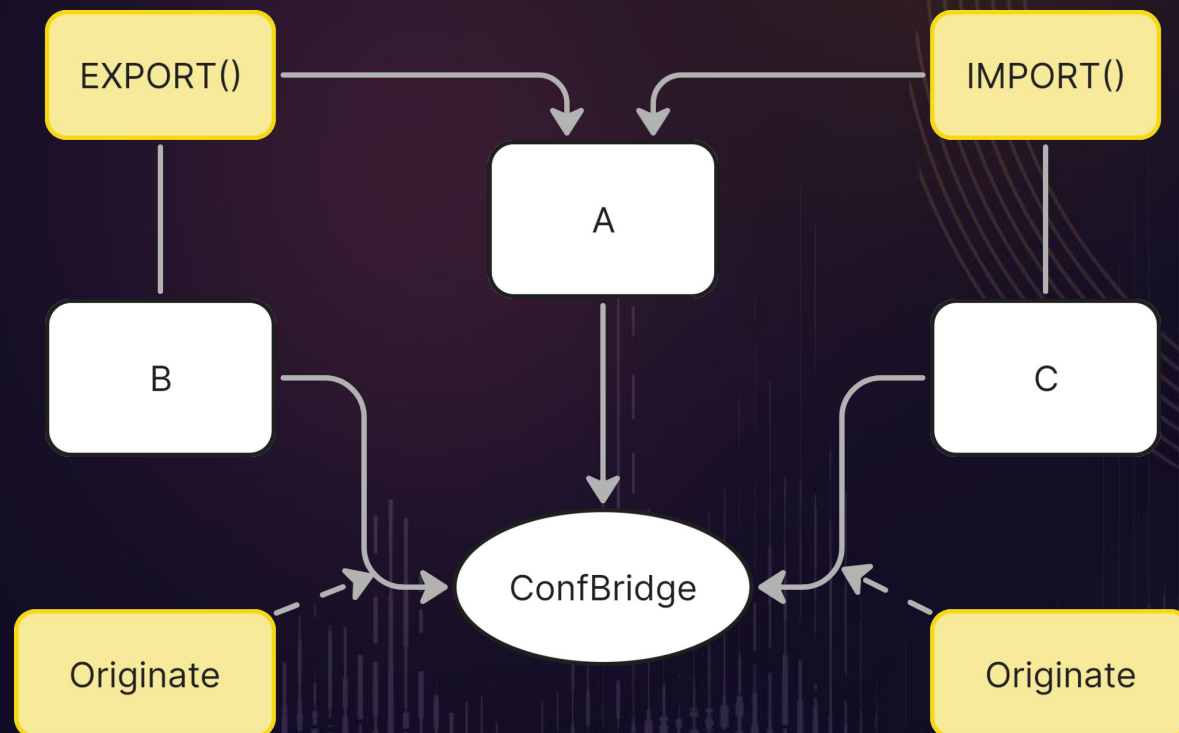
- Работает для Originate
- Работает при оптимизации Local каналов

func_channel.c: static int func_mchan_read(...)

```
mchan = ast_channel_get_by_name(ast_channel_linkedid(chan));
sprintf(template, "${%s}", data); /* SAFE */
ast_str_substitute_variables(buf, len, mchan ? mchan : chan, template);
if (mchan) {
    ast_channel_unref(mchan);
}
return 0;
```

Case 3: собрать данные конференции

- Знаем имя канала
- Знаем участников конференции
- linkedid разные



Ответ: EXPORT/IMPORT

- Прямой доступ по **точному** имени канала
- Работа с функциями при чтении и записи

Ответ: EXPORT/IMPORT

func_export.c: static int func_export_write(...)

```
ochan = ast_channel_get_by_name(args.channel);
if (!ochan) {
    ast_log(LOG_WARNING, "Channel '%s' not found! '%s' not set.\n", args.channel, args.var);
    return -1;
}

pbx_builtin_setvar_helper(ochan, args.var, value);
ast_channel_unref(ochan);
```

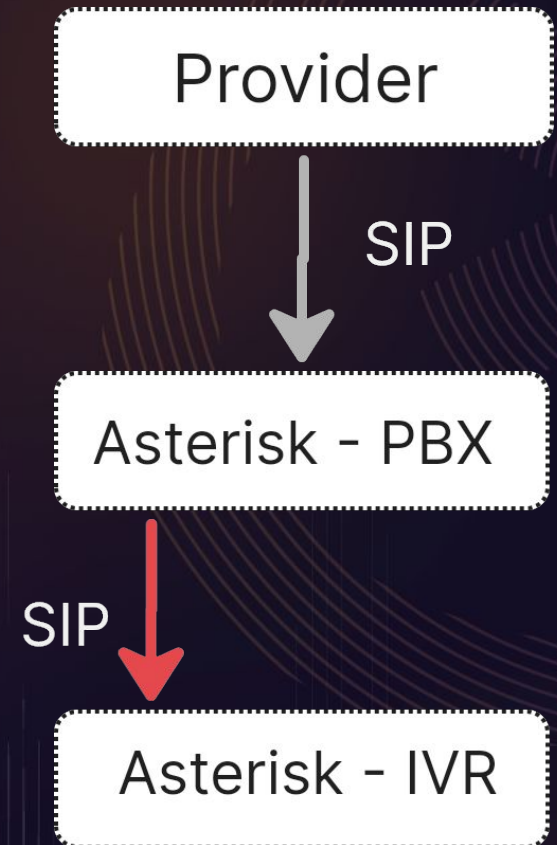
func_export.c: static int func_export_write(...)

```
ochan = ast_channel_get_by_name(args.channel);
if (!ochan) {
    ast_log(LOG_WARNING, "Channel '%s' not found! '%s' not set.\n", args.channel, args.var);
    return -1;
}

pbx_builtin_setvar_helper(ochan, args.var, value);
ast_channel_unref(ochan);
```

Case 4: передать данные на IVR

- Вызов с основного сервера передается на IVR
- Необходимо передать ID клиента



Ответ: PJSIP_HEADER

- Добавление заголовков
- Только в момент создания канала

```
same => n,Dial(PJSIP/s@ivr,,b(initchan,s,1))
```

```
[initchan]
```

```
exten => s,1,NoOp()
```

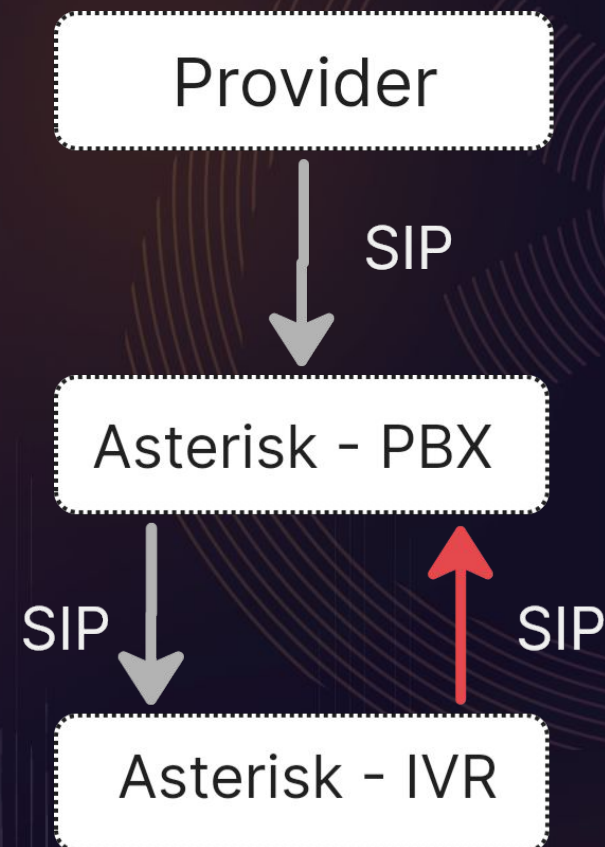
```
same => n,Set(PJSIP_HEADER(add,X-Client-Id)=${ID})
```

```
same => n,Set(PJSIP_HEADER(add,X-Client-Data)=${DATA})
```

```
same => n,Return()
```

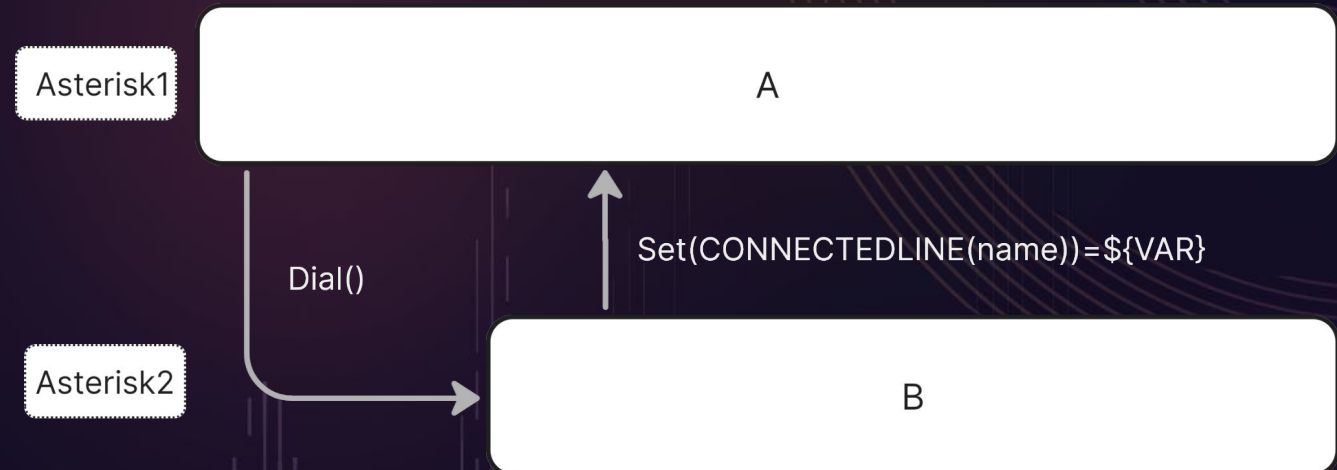
Case 5: передать ответ с IVR сервера

- Вызов с основного сервера передается на IVR
- Один из ответов нужно вернуть на основной сервер



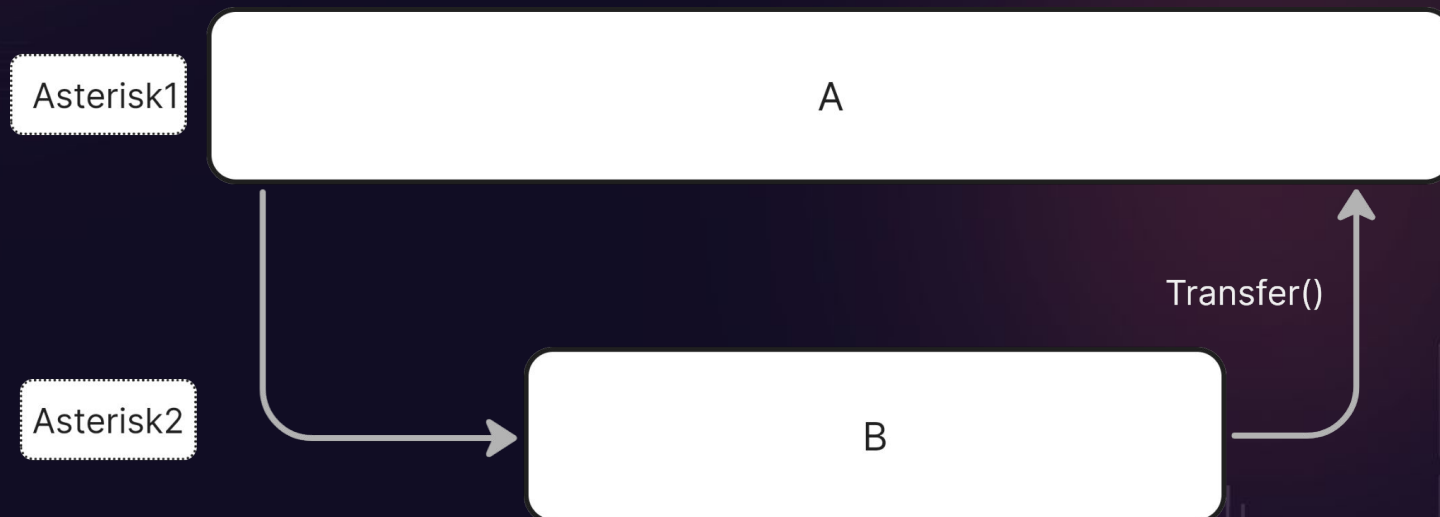
Ответ 5.1: CONNECTED_LINE()

- Если вызов не продолжается
- re-INVITE / UPDATE
- Требует настройки PJSIP
 - `trust_connected_line=yes`
 - `send_connected_line=yes`



Ответ 5.2: Transfer()

- Отправляет REFER
- Можно передать данные в URI

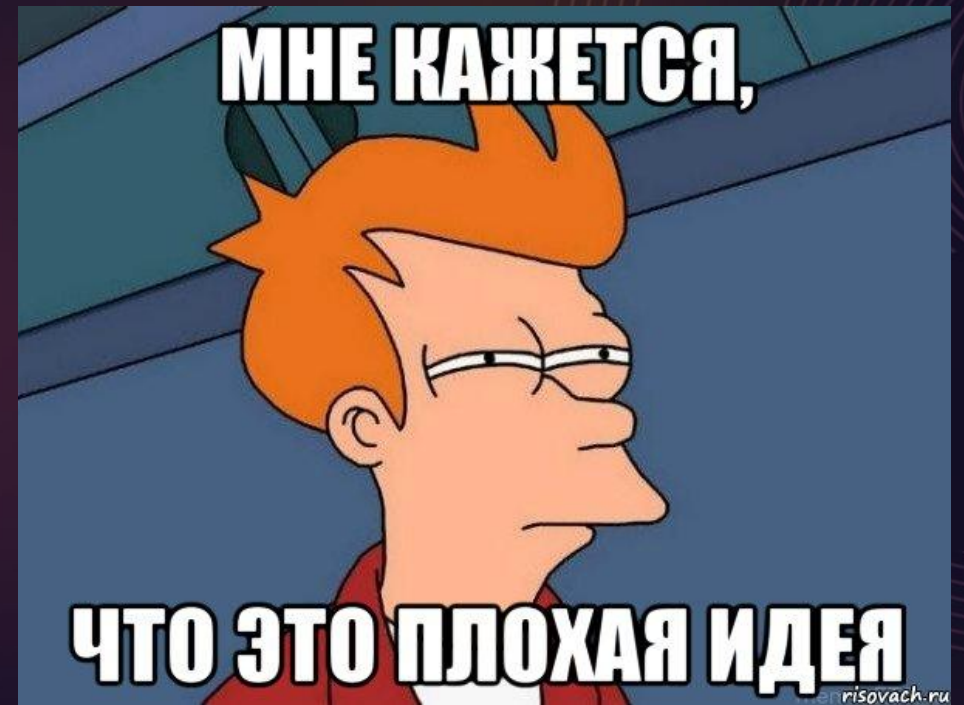


same => n,Transfer(sip:\${number}#\${b>**data**@127.0.0.1)

Bad?

Worse!

Worst...



AstDB

- **SQLITE**

- Запись на диск
- Необходимость очищать данные
- Под нагрузкой задержка записи растёт

- Лучше **ODBC**

```
[chan1]
same => n,Set(DB(Vars/${CALL_ID}/AnswerTime)=${EPOCH})

[chan2]
same => n,Set(BILL=${DB(Vars/${CALL_ID}/AnswerTime)})
same => n,Set(DB_DELETE(vars/${CALL_ID}/AnswerTime)=1)
```

SHARED(varname,channel)

- Отдельное пространство переменных
- Туманное описание / полное отсутствие примеров

« Finally, realize that there is an **inherent race** between channels operating at the same time, fiddling with each others' internal variables, which is why this special variable namespace exists; **it is to remind you** that variables in the SHARED namespace may change at any time, without warning. You **should therefore take special care** to ensure that when using the SHARED namespace, you retrieve the variable and store it in a regular channel variable before using it in a set of calculations (or **you might be surprised** by the result). »

SHARED(varname,channel)

- При установке переменной - **добавление datastore**
- При завершении канала - **удаление**
- Неявный поиск канала по префиксу
- Необходимость использовать LOCK/UNLOCK/TRYLOCK

Простота - залог успеха

SHARED(varname,channel)

```
if (!ast_strlen_zero(args.chan)) {
    char *prefix = ast_alloca(strlen(args.chan) + 2);
    sprintf(prefix, "%s-", args.chan);
    if (!(c_ref = ast_channel_get_by_name(args.chan)) && !(c_ref = ast_channel_get_by_name_prefix(prefix, strlen(prefix)))) {
        ast_log(LOG_ERROR, "Channel '%s' not found! Variable '%s' not set to '%s'.\n", args.chan, args.var, value);
        return -1;
    }
    chan = c_ref;
} else if (!chan) {
    ast_log(LOG_WARNING, "No channel was provided to %s function.\n", cmd);
    return -1;
}

len = 9 + strlen(args.var); /* SHARED() + var */
shared_buffer = ast_malloc(len);
if (!shared_buffer) {
    if (c_ref) {
        ast_channel_unref(c_ref);
    }
    return -1;
}

ast_channel_lock(chan);

if (!(varstore = ast_channel_datastore_find(chan, &shared_variable_info, NULL))) {
    if (!(varstore = ast_datastore_alloc(&shared_variable_info, NULL))) {
        ast_log(LOG_ERROR, "Unable to allocate new datastore. Shared variable not set.\n");
        ast_channel_unlock(chan);
        if (c_ref) {
            c_ref = ast_channel_unref(c_ref);
        }
        return -1;
    }
}

if (!(varshd = ast_calloc(1, sizeof(*varshd)))) {
    ast_log(LOG_ERROR, "Unable to allocate variable structure. Shared variable not set.\n");
    ast_datastore_free(varstore);
    ast_channel_unlock(chan);
}
```

Итоги

- **Передать переменные:**
возможно в любой момент и в любом направлении
- **Между серверами** в том числе
- **Документация** местами туманна и не структурирована
- **Знание кода** полезно для понимания процессов и доступны каждому

Спасибо за внимание!

Буду рад ответить на все ваши вопросы
сейчас или свяжитесь со мной в будущем:

IQTEK

Гончаровский Игорь



igorg@iqtek.ru



@lgorrG

