

Schlumpfen mit OpenSer

Hannes Sowa
&
Frederick Bullik

Entropia e.V.

20. Januar 2008

Inhalt

- 1 Einleitung
 - Was ist Openser
 - Wieso Openser
- 2 Installation
 - Requirements
 - Bau0rn
- 3 Configuration
 - Configuration
 - Test0rn
 - I can has docu?
- 4 Entwicklung
 - Grundgerüst
 - Speicherverwaltung
 - Bau0rn und Test0rn

Inhalt

- 1 **Einleitung**
 - Was ist Openser
 - Wieso Openser
- 2 **Installation**
 - Requirements
 - Bau0rn
- 3 **Configuration**
 - Configuration
 - Test0rn
 - I can has docu?
- 4 **Entwicklung**
 - Grundgerüst
 - Speicherverwaltung
 - Bau0rn und Test0rn

Inhalt

- 1 **Einleitung**
 - Was ist Openser
 - Wieso Openser
- 2 **Installation**
 - Requirements
 - Bau0rn
- 3 **Configuration**
 - Configuration
 - Test0rn
 - I can has docu?
- 4 **Entwicklung**
 - Grundgerüst
 - Speicherverwaltung
 - Bau0rn und Test0rn

Inhalt

- 1 **Einleitung**
 - Was ist Openser
 - Wieso Openser
- 2 **Installation**
 - Requirements
 - Bau0rn
- 3 **Configuration**
 - Configuration
 - Test0rn
 - I can has docu?
- 4 **Entwicklung**
 - Grundgerüst
 - Speicherverwaltung
 - Bau0rn und Test0rn

Gliederung

- 1 Einleitung
 - Was ist Openser
 - Wieso Openser
- 2 Installation
 - Requirements
 - Bau0rn
- 3 Configuration
 - Configuration
 - Test0rn
 - I can has docu?
- 4 Entwicklung
 - Grundgerüst
 - Speicherverwaltung
 - Bau0rn und Test0rn

OpenSer

- Fork von Ser
- Kümmert sich gröstenteils nur um SIP
- Vergleichbar mit Webserver auf SIP-Basis

OpenSer

- Fork von Ser
- Kümmert sich gröstenteils nur um SIP
- Vergleichbar mit Webserver auf SIP-BAsis

OpenSer

- Fork von Ser
- Kümmert sich gröstenteils nur um SIP
- Vergleichbar mit Webserver auf SIP-Basis

Warum?

- TCP support
- Security
- Skalierbar

Warum?

- TCP support
- Security
- Skalierbar

Warum?

- TCP support
- Security
- Skalierbar

Gliederung

- 1 Einleitung
 - Was ist Openser
 - Wieso Openser
- 2 Installation
 - Requirements
 - Bau0rn
- 3 Configuration
 - Configuration
 - Test0rn
 - I can has docu?
- 4 Entwicklung
 - Grundgerüst
 - Speicherverwaltung
 - Bau0rn und Test0rn

Was brauchen Wir?

- Lex
- Bison oder Yacc
- MySql oder Postgress

Was brauchen Wir?

- Lex
- Bison oder Yacc
- MySql oder Postgress

Was brauchen Wir?

- Lex
- Bison oder Yacc
- MySql oder Postgress

Das Makefile

Editire:

- `exclude_modules?=`
- `include_modules?=`

Makefile.dfs

Auskommentieren:

- -DNO_DEBUG
- -DDBG_F_MALLO
- -DDBG_QM_MALLO

Makefile.dfs

Auskommentieren:

- -DNO_DEBUG
- -DDBG_F_MALLO
- -DDBG_QM_MALLO

Makefile.dfs

Auskommentieren:

- -DNO_DEBUG
- -DDBG_F_MALLO
- -DDBG_QM_MALLO

Wo landet was?

/usr/local/sbin

- openser ⇒ Guess what
- openserctl ⇒ OpenSer steuern
- openserdbctl ⇒ Datenbank Anlegen, Verwalten
- openserunix ⇒ Steuern über Unix-Sockets

/usr/local/etc/openser

- dictionary.radius ⇒ Nur für Radius
- openser.cfg.sample ⇒ Sample
- openserctlrc.sample ⇒ Sample
- openser.cfg ⇒ OpenSer Config File
- openserctlrc ⇒ openserdbctl Config File

Wo landet was?

/usr/local/sbin

- `openser` ⇒ Guess what
- `openserctl` ⇒ OpenSer steuern
- `openserdbctl` ⇒ Datenbank Anlegen, Verwalten
- `openserunix` ⇒ Steuern über Unix-Sockets

/usr/local/etc/openser

- `dictionary.radius` ⇒ Nur für Radius
- `openser.cfg.sample` ⇒ Sample
- `openserctlrc.sample` ⇒ Sample
- `openser.cfg` ⇒ OpenSer Config File
- `openserctlrc` ⇒ `openserdbctl` Config File

Wo landet was?

/usr/local/sbin

- `openser` ⇒ Guess what
- `openserctl` ⇒ OpenSer steuern
- `openserdbctl` ⇒ Datenbank Anlegen, Verwalten
- `openserunix` ⇒ Steuern über Unix-Sockets

/usr/local/etc/openser

- `dictionary.radius` ⇒ Nur für Radius
- `openser.cfg.sample` ⇒ Sample
- `openserctrlrc.sample` ⇒ Sample
- `openser.cfg` ⇒ OpenSer Config File
- `openserctrlrc` ⇒ `openserdbctl` Config File

Wo landet was?

/usr/local/sbin

- `openser` ⇒ Guess what
- `openserctl` ⇒ OpenSer steuern
- `openserdbctl` ⇒ Datenbank Anlegen, Verwalten
- `openserunix` ⇒ Steuern über Unix-Sockets

/usr/local/etc/openser

- `dictionary.radius` ⇒ Nur für Radius
- `openser.cfg.sample` ⇒ Sample
- `openserctrlrc.sample` ⇒ Sample
- `openser.cfg` ⇒ OpenSer Config File
- `openserctrlrc` ⇒ `openserdbctl` Config File

Wo landet was?

/usr/local/sbin

- `openser` ⇒ Guess what
- `openserctl` ⇒ OpenSer steuern
- `openserdbctl` ⇒ Datenbank Anlegen, Verwalten
- `openserunix` ⇒ Steuern über Unix-Sockets

/usr/local/etc/openser

- `dictionary.radius` ⇒ Nur für Radius
- `openser.cfg.sample` ⇒ Sample
- `openserctlrc.sample` ⇒ Sample
- `openser.cfg` ⇒ OpenSer Config File
- `openserctlrc` ⇒ `openserdbctl` Config File

Wo landet was?

/usr/local/sbin

- `openser` ⇒ Guess what
- `openserctl` ⇒ OpenSer steuern
- `openserdbctl` ⇒ Datenbank Anlegen, Verwalten
- `openserunix` ⇒ Steuern über Unix-Sockets

/usr/local/etc/openser

- `dictionary.radius` ⇒ Nur für Radius
- `openser.cfg.sample` ⇒ Sample
- `openserctrlrc.sample` ⇒ Sample
- `openser.cfg` ⇒ OpenSer Config File
- `openserctrlrc` ⇒ `openserdbctl` Config File

Wo landet was?

`/usr/local/sbin`

- `openser` ⇒ Guess what
- `openserctl` ⇒ OpenSer steuern
- `openserdbctl` ⇒ Datenbank Anlegen, Verwalten
- `openserunix` ⇒ Steuern über Unix-Sockets

`/usr/local/etc/openser`

- `dictionary.radius` ⇒ Nur für Radius
- `openser.cfg.sample` ⇒ Sample
- `openserctrlrc.sample` ⇒ Sample
- `openser.cfg` ⇒ OpenSer Config File
- `openserctrlrc` ⇒ `openserdbctl` Config File

Wo landet was?

/usr/local/sbin

- `openser` ⇒ Guess what
- `openserctl` ⇒ OpenSer steuern
- `openserdbctl` ⇒ Datenbank Anlegen, Verwalten
- `openserunix` ⇒ Steuern über Unix-Sockets

/usr/local/etc/openser

- `dictionary.radius` ⇒ Nur für Radius
- `openser.cfg.sample` ⇒ Sample
- `openserctrlc.sample` ⇒ Sample
- `openser.cfg` ⇒ OpenSer Config File
- `openserctrlc` ⇒ `openserdbctl` Config File

Wo landet was?

/usr/local/sbin

- `openser` ⇒ Guess what
- `openserctl` ⇒ OpenSer steuern
- `openserdbctl` ⇒ Datenbank Anlegen, Verwalten
- `openserunix` ⇒ Steuern über Unix-Sockets

/usr/local/etc/openser

- `dictionary.radius` ⇒ Nur für Radius
- `openser.cfg.sample` ⇒ Sample
- `openserctrlc.sample` ⇒ Sample
- `openser.cfg` ⇒ OpenSer Config File
- `openserctrlc` ⇒ `openserdbctl` Config File

Wo landet was?

/usr/local/sbin

- `openser` ⇒ Guess what
- `openserctl` ⇒ OpenSer steuern
- `openserdbctl` ⇒ Datenbank Anlegen, Verwalten
- `openserunix` ⇒ Steuern über Unix-Sockets

/usr/local/etc/openser

- `dictionary.radius` ⇒ Nur für Radius
- `openser.cfg.sample` ⇒ Sample
- `openserctrlrc.sample` ⇒ Sample
- `openser.cfg` ⇒ OpenSer Config File
- `openserctrlrc` ⇒ `openserdbctl` Config File

Wo landet was?

/usr/local/sbin

- `openser` ⇒ Guess what
- `openserctl` ⇒ OpenSer steuern
- `openserdbctl` ⇒ Datenbank Anlegen, Verwalten
- `openserunix` ⇒ Steuern über Unix-Sockets

/usr/local/etc/openser

- `dictionary.radius` ⇒ Nur für Radius
- `openser.cfg.sample` ⇒ Sample
- `openserctrlrc.sample` ⇒ Sample
- `openser.cfg` ⇒ OpenSer Config File
- `openserctrlrc` ⇒ `openserdbctl` Config File

Gliederung

- 1 Einleitung
 - Was ist Openser
 - Wieso Openser
- 2 Installation
 - Requirements
 - Bau0rn
- 3 **Configuration**
 - Configuration
 - Test0rn
 - I can has docu?
- 4 Entwicklung
 - Grundgerüst
 - Speicherverwaltung
 - Bau0rn und Test0rn

openser.cfg

Openser.cfg besteht aus 3 Abschnitten:

- Globale Parameter
- Modul Parameter
- Routing Logic

openser.cfg

Openser.cfg besteht aus 3 Abschnitten:

- Globale Parameter
- Modul Parameter
- Routing Logic

openser.cfg

Openser.cfg besteht aus 3 Abschnitten:

- Globale Parameter
- Modul Parameter
- Routing Logic

Globale Parameter

- Wie Verhält sich OpenSer
- Welche Module laden wir.

Globale Parameter

- Wie Verhält sich OpenSer
- Welche Module laden wir.

Beispiel

```
debug=3  
fork=no  
children=4  
listen=127.0.0.1  
port=5060  
mpath=" ./modules/"  
loadmodule "mysql/mysql.so"  
loadmodule "tm/tm.so"  
loadmodule "rr/rr.so"
```

Modul Parameter

- Konfiguration der Module
- `modparam("<Modulname >", "<Key >", "<Wert >")`

Modul Parameter

- Konfiguration der Module
- `modparam("<Modulname >", "<Key >", "<Wert >")`

Beispiel

```
modparam("usrloc", "db_mode", 3)  
modparam("usrloc", "db_url",  
"mysql://openser:openser@localhost/openser")  
modparam("auth_db", "calculate_ha1", 1)  
modparam("auth_db", "password_column", "password")
```

Routing Logic I

- 3 Routen:
 - Default
 - R-Rohr
 - Reply
- Blöcke: `route[1]{}`, `route(1)`
- `if()`

Routing Logic I

- 3 Routen:
 - Default
 - R-Rohr
 - Reply
- Blöcke: `route[1]{}`, `route(1)`
- `if()`

Routing Logic I

- 3 Routen:
 - Default
 - R-Rohr
 - Reply
- Blöcke: `route[1]{}`, `route(1)`
- `if()`

Routing Logic I

- 3 Routen:
 - Default
 - R-Rohr
 - Reply
- Blöcke: `route[1]{}`, `route(1)`
- `if()`

Routing Logic I

- 3 Routen:
 - Default
 - R-Rohr
 - Reply
- Blöcke: `route[1]{}`, `route(1)`
- `if()`

Routing Logic I

- 3 Routen:
 - Default
 - R-Rohr
 - Reply
- Blöcke: `route[1]{}`, `route(1)`
- `if()`

Beispiel

```
route{
if (!mf_process_maxfwd_header("10")) {
    sl_send_reply("483", "Too Many Hops");
    exit;
};

if(method==INVITE){
    if(!lookup ("location")){
        sl_send_reply("404", "Not Found");
        exit;
    }
}

route(1);
exit();
}
```

sipp

- Server: `./sipp -sn uas`
- Client: `./sipp -sn uac 127.0.0.1`
- XML config

sipp

- Server: `./sipp -sn uas`
- Client: `./sipp -sn uac 127.0.0.1`
- XML config

sipp

- Server: `./sipp -sn uas`
- Client: `./sipp -sn uac 127.0.0.1`
- XML config

Compil0rn

Dependencies

Docbook-Umgebung

Module

- make modules-docbook
- make modules-readme

Datenbank

make dbschema-docbook

⇒ ./doc/database/

Compil0rn

Dependencies

Docbook-Umgebung

Module

- make modules-docbook
- make modules-readme

Datenbank

make dbschema-docbook

⇒ ./doc/database/

Compil0rn

Dependencies

Docbook-Umgebung

Module

- make modules-docbook
- make modules-readme

Datenbank

make dbschema-docbook

⇒ ./doc/database/

Compil0rn

Dependencies

Docbook-Umgebung

Module

- make modules-docbook
- make modules-readme

Datenbank

make dbschema-docbook

⇒ ./doc/database/

Compil0rn

Dependencies

Docbook-Umgebung

Module

- make modules-docbook
- make modules-readme

Datenbank

make dbschema-docbook

⇒ ./doc/database/

I haz the internetz

- <http://www.openser.org/dokuwiki/doku.php>
- <http://www.openser.org/mos/view/OpenSER-Documentation-Repository/>

I haz the internetz

- <http://www.openser.org/dokuwiki/doku.php>
- <http://www.openser.org/mos/view/OpenSER-Documentation-Repository/>

Gliederung

- 1 Einleitung
 - Was ist Openser
 - Wieso Openser
- 2 Installation
 - Requirements
 - Bau0rn
- 3 Configuration
 - Configuration
 - Test0rn
 - I can has docu?
- 4 Entwicklung
 - Grundgerüst
 - Speicherverwaltung
 - Bau0rn und Test0rn

Includes

```
#include "../mem/shm_mem.h"  
#include "../sr_module.h"  
#include "../mem/mem.h"  
#include "../error.h"  
#include "../pvar.h"
```

cmd_export_t

```
static cmd_export_t cmds[]={
    {"printa",printa,2,printa_fixup,0,
     REQUEST_ROUTE | FAILURE_ROUTE |
     ONREPLY_ROUTE | BRANCH_ROUTE | ERROR_ROUTE},
    {0, 0, 0, 0, 0, 0}
};
```

param_export_t

```
static param_export_t params[]={  
    {"wordlist",STR_PARAM,&wordlst},  
    {"buffer_size",INT_PARAM,&buf_size},  
    {0,0,0}  
};
```

module_exports

```

struct module_exports exports = {
    "analog",
    DEFAULT_DLFLAGS, /* dlopen flags */
    cmds,             /* exported functions */
    params,           /* exported parameters */
    0,                /* exported statistics */
    0,                /* exported MI functions */
    0,                /* exported pseudo-variables */
    0,                /* extra processes */
    mod_init,         /* module initialization function */
    0,                /* response function */
    mod_destroy,      /* destroy function */
    0                 /* per-child init function */
};
  
```


Speicher Verwaltung

pkg

“Lokaler „Speicher

- pkg_malloc(<Size >)
- pkg_free(<Variable >)

shm

”Globaler Speicher”

- shm_malloc(<Size >)
- shm_free(<Variable >)

Speicher Verwaltung

pkg

“Lokaler „Speicher

- pkg_malloc(<Size >)
- pkg_free(<Variable >)

shm

”Globaler Speicher”

- shm_malloc(<Size >)
- shm_free(<Variable >)

Speicher Verwaltung

pkg

“Lokaler „Speicher

- pkg_malloc(<Size >)
- pkg_free(<Variable >)

shm

”Globaler Speicher”

- shm_malloc(<Size >)
- shm_free(<Variable >)

Speicher Verwaltung

pkg

“Lokaler „Speicher

- pkg_malloc(<Size >)
- pkg_free(<Variable >)

shm

”Globaler Speicher”

- shm_malloc(<Size >)
- shm_free(<Variable >)

Speicher Verwaltung

pkg

“Lokaler „Speicher

- pkg_malloc(<Size >)
- pkg_free(<Variable >)

shm

”Globaler Speicher”

- shm_malloc(<Size >)
- shm_free(<Variable >)

Speicher Verwaltung

pkg

“Lokaler „Speicher

- pkg_malloc(<Size >)
- pkg_free(<Variable >)

shm

”Globaler Speicher”

- shm_malloc(<Size >)
- shm_free(<Variable >)

Bauen

- make im Verzeichnis
- make im root

Bauen

- make im Verzeichnis
- make im root

Debug0rn

- debug=3
- fork=no
- log_stderr=yes
- LM_INFO(" <String >")
- LM_ERR(" <String >")

Debug0rn

- debug=3
- fork=no
- log_stderr=yes
- LM_INFO(" <String >")
- LM_ERR(" <String >")

Debug0rn

- debug=3
- fork=no
- log_stderr=yes
- LM_INFO(" <String >")
- LM_ERR(" <String >")

Debug0rn

- debug=3
- fork=no
- log_stderr=yes
- LM_INFO(" <String >")
- LM_ERR(" <String >")

Debug0rn

- debug=3
- fork=no
- log_stderr=yes
- LM_INFO(" <String >")
- LM_ERR(" <String >")

Fragen ?

Noch Fragen?

The End

The End