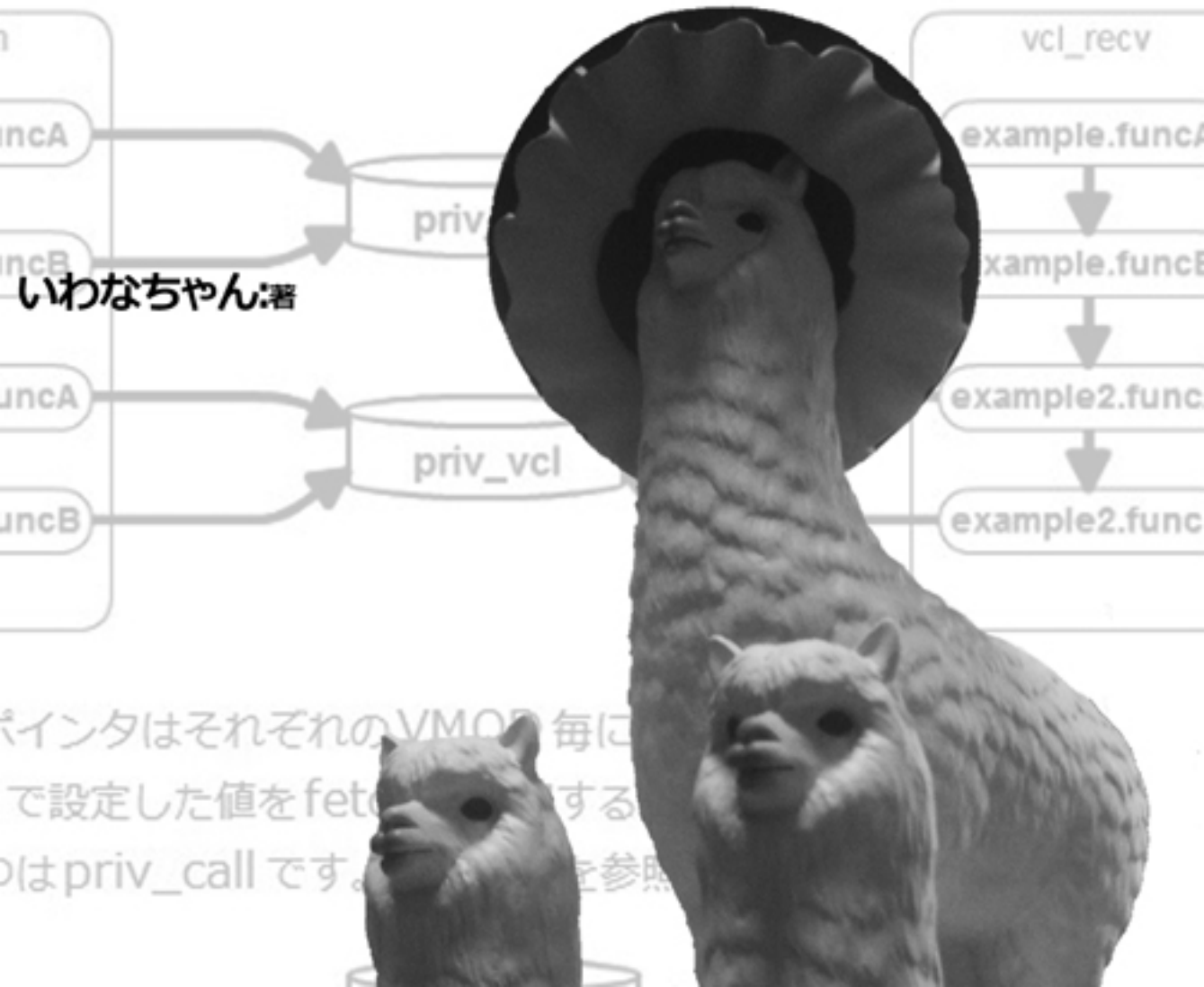
reverseproxy

Varnish Cache

inline-C/VMOD GUIDEBOOK

inline-C/VMOD ガイドブック

いわなちゃん著



目次

またインラインCで扱った関数と同様に、必ず第一引数はspとなります。

```
int vmod_len(struct sess *sp, const char *p){
        return(strlen(p));
}
```

第二引数以降は受け取る変数によって変わってきます。
init_functionについては後述します。

*VMOD*で使用可能な変数の型

VMODで利用可能な型はほぼVCLと同じです。しかし一部非推奨だっり特殊な型があったりします。

VCLでの型名	Cでの型名	戻値	引数	備考
BACKEND	struct director *	○	○	記述なし
BOOL	unsigned	○	○	非推奨
DURATION	double	○	○	
INT	int	○	○	
IP	struct sockaddr_storage *	△	○	非推奨
STRING	const char *	○	○	
TIME	double	○	○	
HEADER	enum gethdr_e, const char *	×	○	非推奨
REAL	double	○	○	
STRING_LIST	const char *, ...	×	○	
PRIV_VCL	struct vmod_priv *	△	○	
PRIV_CALL	struct vmod_priv *	△	○	
VOID	void	○	×	

戻値が△なのは、書き込みできる変数が存在せず、使い道がさほど思いつかなかった為です。また非推奨のものは公式のドキュメントに記載されていたものです。
それぞれの変数を引数・戻値に持つ簡単な関数を作って解説します。

BACKEND

バックエンドの情報を格納しています。引数、戻り値共に指定可能です。

■vcc

```
Function BACKEND tbackend(BACKEND)
```

■c

```
struct director * vmod_tbackend(struct sess *sp, struct director *p){
        return p;
}
```

■VCL

```
set req.backend = example.tbackend(req.backend);
```

もしdirectorのメンバーにアクセスする場合は以下のヘッダをincludeする必要があります。

```
#include "bin/varnishd/cache.h"
#include "bin/varnishd/cache_backend.h"
```

例として指定したバックエンドが正常の場合はそれを返却し、そうでない場合は現在選択されているバックエンドを返却します。

```
■vcc
Function BACKEND gethealthydirector(BACKEND)

■c
struct director * vmod_gethealthydirector(
struct sess *sp, struct director *p){
        if(VDI_Healthy(p, sp)){
                return p;
        }
        return sp->director;
}

■VCL
set req.backend = example.gethealthydirector(client_2);
```

VDI_Healthyは、バックエンドの状態を返却します。
利用するには以下のヘッダをincludeする必要があります。

```
#include "bin/varnishd/cache.h"
```

BOOL

真偽が入っている型です。

```
■vcc
Function BOOL tbool(BOOL)

■c
unsigned vmod_tbool(struct sess *sp, unsigned p){
        return p;
}

■VCL
set req.esi = example.tbool(req.esi);
```

inline-C/VMOD ガイドブック