

Création de l'exemple SplitMsg

Allez dans un nouveau répertoire pour y créer des fichiers.

Fichier de description DSX

Pour celui-ci, choisissez vous-même un nom explicite. Ce fichier est le **fichier de description DSX** et sera nommé comme tel par la suite.

Collez dedans le texte suivant:

```
#!/usr/bin/env python

from dsx import *

# Partie 1
producer_model = TaskModel(
    'producer',
    outfifos = ['output'],
    impl = [
        SwTask('prod_func',
               stack_size = 128,
               sources = ['producer.c'])
    ] )

consumer_model = TaskModel(
    'consumer',
    outfifos = ['input'],
    impl = [
        SwTask('cons_func',
               stack_size = 128,
               sources = ['consumer.c'])
    ] )

# Partie 2
fifo0 = Mwmr('fifo0', 4, 4)

tcg = Tcg(
    Task('prod0', producer_model,
         {'output':fifo0} ),
    Task('cons0', consumer_model,
         {'input':fifo0} ),
)

# Partie 3
posix = Posix()
tcg.generate(posix)
TopMakefile(posix)
```

Important: La ligne `#!/usr/bin/env python` doit être la *première ligne* du fichier.

Rendez ce fichier exécutable

```
$ chmod +x le_nom_de_fichier
```

Fichier producer.c

Dans `producer.c`, mettez:

```

#include <srl.h>
#include "producer_proto.h"

FUNC(prod_func)
{
    srl_mwmr_t output = GET_ARG(output);
    char buf[32] = "!dlrow olleH";

    srl_log_printf(NONE, "Sending \"%s\"\n", buf);
    srl_mwmr_write(output, buf, 8);
}

```

Fichier consumer.c

Dans `consumer.c`, mettez:

```

#include <srl.h>
#include "consumer_proto.h"

FUNC(cons_func)
{
    srl_mwmr_t input = GET_ARG(input);
    char buf_in[32];
    char buf_out[32];
    int i, last_char;

    srl_mwmr_read(input, buf_in, 8);
    srl_log_printf(NONE, "Received \"%s\"\n", buf_in);

    for ( last_char=0; buf_in[last_char]; ++last_char )
        ;
    for ( i = 0; i<last_char; ++i )
        buf_out[i] = buf_in[last_char-1-i];
    buf_out[last_char] = '\0';

    srl_log_printf(NONE, "More readable when \"%s\"\n\n", buf_out);
}

```