

# 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 = 256,
               sources = ['producer.c'])
    ] )

consumer_model = TaskModel(
    'consumer',
    infifos = ['input'],
    impl = [
        SwTask('cons_func',
               stack_size = 256,
               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] = "...World";
    srl_log_printf(NONE, "Producer : Hello...\n");
    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[32];
    srl_mwmr_read(input, buf, 8);
    srl_log_printf(NONE, "Consumer : %s\n\n", buf);
}
```