

Création de l'exemple SplitMsg

Allez dans un nouveau répertoire pour y créer des fichiers.

Description des deux tâches

Dans le nouveau répertoire, créez un répertoire 'src' et allez dedans.

Créez un fichier `producer.task` contenant:

```
TaskModel(  
    'producer',  
    ports = { 'output' : MwmrOutput(32) },  
    impls = [  
        SwTask('prod_func',  
              stack_size = 2048,  
              sources = ['producer.c'])  
    ] )
```

Créez un fichier `consumer.task` contenant:

```
TaskModel(  
    'consumer',  
    ports = { 'input' : MwmrInput(32) },  
    impls = [  
        SwTask('cons_func',  
              stack_size = 2048,  
              sources = ['consumer.c'])  
    ] )
```

Implémentation des deux tâches

Toujours dans 'src',

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, 32);  
}
```

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, 32);  
    srl_log_printf(NONE, "Consumer : %s\n\n", buf);  
}
```

```
}
```

On vient de créer deux modèles de tâches (décrits dans les fichiers `.task`, et leurs implémentations.

Fichier de description DSX/L

Revenez dans le répertoire de l'application (donc le répertoire parent de `src`).

Créez un fichier de description pour 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

import dsx

# Partie 1 : définition du TCG (Graphe des Tâches et des Communications)

fifo0 = dsx.Mwmmr('fifo0', 32, 4)

tcg = dsx.Tcg(
    dsx.Task('prod0', 'producer',
             {'output':fifo0} ),
    dsx.Task('cons0', 'consumer',
             {'input':fifo0} ),
)

# Partie 2 : génération du code exécutable sur station de travail POSIX

tcg.generate(dsx.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
```