

```

/***** A compiler avec

gcc -o Exemple1_await_generate Exemple1_await_generate.c \
    traceinstantsf.c -I $CHEMIN/ft_v1.1/include -L $CHEMIN/ft_v1.1/lib \
    -lftthread -lpthread
*****/

#include "ftthread.h"
#include "stdio.h"
#include "unistd.h"
#include "traceinstantsf.h"
#include "stdlib.h"
#include "pthread.h"

ft_thread_t    ft_trace, ft_broadcaster;
ft_thread_t    ft_awaiter[3];
ft_scheduler_t sched;
ft_event_t     evt;

void awaiter (void *arg)
{
    long i, n, res;

    for (i = 0; i < 5; i++) {
        fprintf(stdout, "awater%d en attente d'un evenement.\n", (long)arg);
        res = ft_thread_await(evt);

        if (res == OK) {
            ft_thread_get_value(evt, 0, (void *)&n);
            fprintf(stdout,
                "awaiter%d a reçu l'évenement numero %d.\n",
                (long)arg, n);
        }
        ft_thread_cooperate();
    }
}

void broadcaster (void *arg)
{
    long i, res;

    for (i=0;; ++i) {
        fprintf(stdout, "broadcaster broadcast l'évenement numero %d.\n", i);
        res = ft_scheduler_broadcast_value(evt, (void *)i);
        if (res != OK) {
            fprintf(stdout, "ERREUR.\n", res);
            fprintf(stdout, "On arrete tout.\n");
            exit(0);
        }
        ft_thread_cooperate();
    }
}

void join_waiters (void *arg)
{
    long i;

    fprintf(stdout, "Debut de join_waiters.\n");

    for (i = 0; i < 3; ++i) {
        ft_thread_join(ft_awaiter[i]);
    }
}

```

```

}

fprintf(stdout, "Fin de tous les awaiter.\n");

fprintf(stdout, "Stoper la trace.\n");
ft_scheduler_stop(ft_trace);

fprintf(stdout, "Stoper le generateur.\n", (long)arg);
ft_scheduler_stop(ft_broadcaster);

fprintf(stdout, "***** exit(0) *****\n");
exit(0);
}

int main(int argc, char *argv[])
{
    long i;

    sched = ft_scheduler_create ();
    evt = ft_event_create(sched);
    ft_trace = ft_thread_create(sched, traceinstants, NULL, (void *)50);
    for (i = 0; i < 3; ++i) {
        ft_awaiter[i] = ft_thread_create(sched, awaiter, NULL, (void *) (i + 1));
    }
    ft_thread_create(sched, join_waiters, NULL, NULL);
    ft_scheduler_start(sched);

    sched = ft_scheduler_create (); /* un nouveau scheduler */
    ft_broadcaster = ft_thread_create(sched, broadcaster, NULL, NULL);
    ft_scheduler_start(sched);

    fprintf(stdout, "\n\nC'est fini pour le programme principal.\n");

    ft_exit(); /* Bloquant donc. */

    return 0; /* Never reached.*/
    /* Juste pour calmer le compilo pur avoir écrit int main() */
}

/*
$ Exemple2_await_broadcast
>>>>>>>>> instant 0 :
awater1 en attente d'un evenement.

C'est fini pour le programme principal.
awater2 en attente d'un evenement.
awater3 en attente d'un evenement.
broadcaster broadcast l'évenement numero 0.
Debut de join_waiters.
broadcaster broadcast l'évenement numero 1.
>>>>>>>>> instant 1 :
awaiter1 a reçu l'évenement numero 0.
awaiter2 a reçu l'évenement numero 0.
broadcaster broadcast l'évenement numero 2.
awaiter3 a reçu l'évenement numero 0.
broadcaster broadcast l'évenement numero 3.
>>>>>>>>> instant 2 :
awater1 en attente d'un evenement.
awaiter1 a reçu l'évenement numero 1.

```

```
broadcaster broadcast l'evenement numero 4.
awater2 en attente d'un evenement.
awaiter2 a reçu l'evenement numero 1.
broadcaster broadcast l'evenement numero 5.
awater3 en attente d'un evenement.
awaiter3 a reçu l'evenement numero 1.
broadcaster broadcast l'evenement numero 6.
>>>>>>>>> instant 3 :
broadcaster broadcast l'evenement numero 7.
awater1 en attente d'un evenement.
awaiter1 a reçu l'evenement numero 3.
broadcaster broadcast l'evenement numero 8.
awater2 en attente d'un evenement.
awaiter2 a reçu l'evenement numero 3.
broadcaster broadcast l'evenement numero 9.
awater3 en attente d'un evenement.
awaiter3 a reçu l'evenement numero 3.
broadcaster broadcast l'evenement numero 10.
>>>>>>>>> instant 4 :
broadcaster broadcast l'evenement numero 11.
awater1 en attente d'un evenement.
awaiter1 a reçu l'evenement numero 6.
broadcaster broadcast l'evenement numero 12.
awater2 en attente d'un evenement.
awaiter2 a reçu l'evenement numero 6.
broadcaster broadcast l'evenement numero 13.
awater3 en attente d'un evenement.
awaiter3 a reçu l'evenement numero 6.
broadcaster broadcast l'evenement numero 14.
>>>>>>>>> instant 5 :
broadcaster broadcast l'evenement numero 15.
awater1 en attente d'un evenement.
awaiter1 a reçu l'evenement numero 10.
broadcaster broadcast l'evenement numero 16.
awater2 en attente d'un evenement.
awaiter2 a reçu l'evenement numero 10.
broadcaster broadcast l'evenement numero 17.
awater3 en attente d'un evenement.
awaiter3 a reçu l'evenement numero 10.
broadcaster broadcast l'evenement numero 18.
>>>>>>>>> instant 6 :
broadcaster broadcast l'evenement numero 19.
broadcaster broadcast l'evenement numero 20.
broadcaster broadcast l'evenement numero 21.
broadcaster broadcast l'evenement numero 22.
broadcaster broadcast l'evenement numero 23.
Fin de tous les awaiter.
Stoper la trace.
Stoper le generateur.
***** exit(0) *****
broadcaster broadcast l'evenement numero 24
*/
```