

```

/***** 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 "fthread.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);
        }
    }
}

void broadcaster (void *arg)
{
    long i;

    for (i=0;; ++i) {
        fprintf(stdout, "broadcaster broadcast l'évenement numero %d.\n", i);
        ft_scheduler_broadcast_value(evt, (void *)i);
        ft_thread_cooperate();
    }
}

void join_awaiters (void *arg)
{
    long i;

    fprintf(stdout, "Debut de join_awaiters.\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);
    }

    ft_broadcaster = ft_thread_create(sched, broadcaster, NULL, NULL);

    ft_thread_create(sched, join_awaiters, 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() */
}

/*
$ Exemple1_await_broadcast

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

```

```
awater2 en attente d'un evenement.
awaiter2 a reçu l'évenement numero 0.
awater2 en attente d'un evenement.
awaiter2 a reçu l'évenement numero 0.
awater2 en attente d'un evenement.
awaiter2 a reçu l'évenement numero 0.
awaiter3 a reçu l'évenement numero 0.
awater3 en attente d'un evenement.
awaiter3 a reçu l'évenement numero 0.
awater3 en attente d'un evenement.
awaiter3 a reçu l'évenement numero 0.
awater3 en attente d'un evenement.
awaiter3 a reçu l'évenement numero 0.
awater3 en attente d'un evenement.
awaiter3 a reçu l'évenement numero 0.
awater3 en attente d'un evenement.
awaiter3 a reçu l'évenement numero 0.
broadcastor broadcast l'évenement numero 1.
Fin de tous les awaiter.
Stoper la trace.
Stoper le generateur.
***** exit(0) *****
*/
```