```
/***** 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 \
       −lfthread −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_broadcastor;
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 recu l’evenement numero %d.\n",
              (long)arg, n);
    }
    ft_thread_cooperate();
  }
}


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

  ft_thread_unlink();   /* tourne en pthread normal des l’entree */

  for (i=0;; ++i) {
    fprintf(stdout, "broadcastor broadcast l’evenement 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_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_broadcastor);

  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_awaiters, NULL, NULL);
  ft_scheduler_start(sched);

  sched          = ft_scheduler_create (); /* un nouveau scheduler */
  ft_broadcastor = ft_thread_create(sched, broadcastor, 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 ecrit int main() */
}





/*
$ Exemple3_await_broadcast
>>>>>>>>>>> instant 0 :


C’est fini pour le programme principal.
awater1 en attente d’un evenement.
awater2 en attente d’un evenement.
broadcastor broadcast l’evenement numero 0.
broadcastor broadcast l’evenement numero 1.
...
broadcastor broadcast l’evenement numero 6.
awater3 en attente d’un evenement.
broadcastor broadcast l’evenement numero 7.
...
broadcastor broadcast l’evenement numero 12.
Debut de join_awaiters.
broadcastor broadcast l’evenement numero 13.
...
```

```
broadcastor broadcast l’evenement numero 16.
>>>>>>>>>>> instant 1 :
broadcastor broadcast l’evenement numero 17.
...
broadcastor broadcast l’evenement numero 27.
awaiter1 a recu l’evenement numero 0.
broadcastor broadcast l’evenement numero 28.
...
broadcastor broadcast l’evenement numero 34.
awaiter2 a recu l’evenement numero 0.
broadcastor broadcast l’evenement numero 35.
...
broadcastor broadcast l’evenement numero 70.
awaiter3 a recu l’evenement numero 0.
broadcastor broadcast l’evenement numero 71.
...
broadcastor broadcast l’evenement numero 74.
>>>>>>>>>>> instant 2 :
broadcastor broadcast l’evenement numero 75.
awater1 en attente d’un evenement.
awaiter1 a recu l’evenement numero 13.
broadcastor broadcast l’evenement numero 76.
...
broadcastor broadcast l’evenement numero 94.
awater2 en attente d’un evenement.
awaiter2 a recu l’evenement numero 13.
broadcastor broadcast l’evenement numero 95.
awater3 en attente d’un evenement.
awaiter3 a recu l’evenement numero 13.
broadcastor broadcast l’evenement numero 96.
broadcastor broadcast l’evenement numero 97.
broadcastor broadcast l’evenement numero 98.
broadcastor broadcast l’evenement numero 99.
broadcastor broadcast l’evenement numero 100.
>>>>>>>>>>> instant 3 :
broadcastor broadcast l’evenement numero 101.
awater1 en attente d’un evenement.
awaiter1 a recu l’evenement numero 71.
broadcastor broadcast l’evenement numero 102.
awater2 en attente d’un evenement.
awaiter2 a recu l’evenement numero 71.
broadcastor broadcast l’evenement numero 103.
awater3 en attente d’un evenement.
awaiter3 a recu l’evenement numero 71.
broadcastor broadcast l’evenement numero 104.
broadcastor broadcast l’evenement numero 105.
>>>>>>>>>>> instant 4 :
broadcastor broadcast l’evenement numero 106.
awater1 en attente d’un evenement.
awaiter1 a recu l’evenement numero 97.
broadcastor broadcast l’evenement numero 107.
awater2 en attente d’un evenement.
awaiter2 a recu l’evenement numero 97.
broadcastor broadcast l’evenement numero 108.
awater3 en attente d’un evenement.
awaiter3 a recu l’evenement numero 97.
broadcastor broadcast l’evenement numero 109.
...
broadcastor broadcast l’evenement numero 114.
>>>>>>>>>>> instant 5 :
broadcastor broadcast l’evenement numero 115.
awater1 en attente d’un evenement.
awaiter1 a recu l’evenement numero 104.
broadcastor broadcast l’evenement numero 116.
awater2 en attente d’un evenement.
awaiter2 a recu l’evenement numero 104.
```

```
broadcastor broadcast l’evenement numero 117.
awater3 en attente d’un evenement.
awaiter3 a recu l’evenement numero 104.
broadcastor broadcast l’evenement numero 118.
...
broadcastor broadcast l’evenement numero 122.
>>>>>>>>>>> instant 6 :
broadcastor broadcast l’evenement numero 123.
...
broadcastor broadcast l’evenement numero 128.
Fin de tous les awaiter.
Stoper la trace.
broadcastor broadcast l’evenement numero 129.
Stoper le generateur.
********** exit(0) **********
broadcastor broadcast l’evenement numero 130.
...
broadcastor broadcast l’evenement numero 132.
*/
```