

jeu. 02 f'vr. 2017 19:37:22 CEExemple_erreur_await_generate.cPage 1	jeu. 02 f'vr. 2017 19:37:22 CEExemple_erreur_await_generate.cPage 2
<pre> /***** A compiler avec  gcc -o Exemple_erreur_await_generate Exemple_erreur_await_generate.c \ traceinstantsf.c -I \$CHEMIN/ft_v1.1/include -L \$CHEMIN/ft_v1.1/lib \ -lpthread -lpthread  evt_1 est cree dans sched_1, donc lie a sched_1. ft_generator_2 est dans le sched_2, donc linke a sched_2. il tente de generer evt_1.  Resultat :  Dans generator() de ft_generator_2, ft_thread_generate_value(evt_1, (void *)i) retourne la valeur EBADLINK(4) differente de OK(0).  *****/  #include "pthread.h" #include "stdio.h" #include "unistd.h" #include "traceinstantsf.h" #include "stdlib.h" #include "pthread.h"  ft_thread_t    ft_trace_1, ft_generator_2; ft_thread_t    ft_awaiter_1[3]; ft_scheduler_t sched_1, sched_2; ft_event_t     evt_1;  void awaiter (void *arg) {     long i, n, res;      for (i = 0; i &lt; 5; i++) {         fprintf(stdout, "awater%d en attente d'un evenement.\n", (long)arg);         res = ft_thread_await(evt_1);          if (res == OK) {             ft_thread_get_value(evt_1, 0, (void *)&amp;n);             fprintf(stdout,                 "awaiter%d a reçu l'evenement numero %d.\n",                 (long)arg, n);         }     } }  void generator (void *arg) {     long i, res;      for (i=0;; ++i) {         fprintf(stdout, "generator genere l'evenement numero %d.\n", i);          res = ft_thread_generate_value(evt_1, (void *)i);          if (res != OK) {             fprintf(stdout, "ERREUR ==&gt; EBADLINK(%d).\n", res);             fprintf(stdout, "On arrete tout.\n");             exit(0);         }     } } </pre>	<pre> fprintf(stdout, "apres generate.\n"); ft_thread_cooperate(); }  int main(int argc, char *argv[]) {     long i;      sched_1 = ft_scheduler_create ();     sched_2 = ft_scheduler_create ();      evt_1 = ft_event_create(sched_1);      ft_trace_1 = ft_thread_create(sched_1, traceinstants, NULL, (void *)50);      for (i = 0; i &lt; 3; ++i) {         ft_awaiter_1[i] = ft_thread_create(sched_1, awaiter, NULL, (void *)(i + 1));     }      ft_generator_2 = ft_thread_create(sched_2, generator, NULL, NULL);      ft_scheduler_start(sched_1);     ft_scheduler_start(sched_2);      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() */ }  /* \$ Exemple_erreur_await_generate  C'est fini pour le programme principal. &gt;&gt;&gt;&gt;&gt;&gt;&gt;&gt;&gt;&gt; instant 0 : awater1 en attente d'un evenement. generator genere l'evenement numero 0. ERREUR ==&gt; EBADLINK(4). On arrete tout. awater2 en attente d'un evenement. */ </pre>