```

module GreenThreads =
  struct
    type res =
      | Yield of (unit -> res)
      | Fork of (unit -> unit) * (unit ->res)
      | Done;;

    let prompt0 = Delimcc.new_prompt ();;

    let scheduler proc_init =
      let queue = Queue.create () in
      let rec handler result =
        match result with
        | Done         -> if Queue.is_empty queue then () else handler (Queue.pop queue ())
        | Yield k      -> Queue.push k queue; handler (Queue.pop queue ())
        | Fork (p, k) -> Queue.push k queue; run p
      and run proc =
        handler (Delimcc.push_prompt prompt0 (fun () -> proc (); Done))
      in run proc_init;;

    let yield () = Delimcc.shift prompt0 (fun k -> Yield k);;
    let fork proc = Delimcc.shift prompt0 (fun k -> Fork (proc, k));;
    let exit () = Delimcc.shift prompt0 (fun _ -> Done);;
  end

module type Channel =
  sig
    val create : unit -> ('a -> unit) * (unit -> 'a)
  end

module GTChannel : Channel =
  struct
    let create () =
      let queue = Queue.create () in
      (fun v -> (Queue.push v queue; GreenThreads.yield ())),
      (let rec loop () =
         if not (Queue.is_empty queue)
         then Queue.pop queue
         else (GreenThreads.yield (); loop ())
       in loop);;
  end

let ping_pong () =
  GreenThreads.(
    let proc () =
      begin
        fork (fun () -> for i = 1 to 10 do Format.printf "ping !@ "; yield () done);
        fork (fun () -> for i = 1 to 10 do Format.printf "pong !@ "; yield () done);
        exit ()
      end
    in scheduler proc
  )

let sieve () =
  let rec filter reader =
  GreenThreads.(
    let v0 = reader () in
    if v0 = -1 then exit () else
    Format.printf "%d@." v0;
    yield ();
    let (writer', reader') = GTChannel.create () in
    fork (fun () -> filter reader');
    while true
    do
      let v = reader () in
      yield ();
      if v mod v0 <> 0 then writer' v;
      if v = -1 then exit ()
    done
  ) in
let main () =
  GreenThreads.(
    let (writer, reader) = GTChannel.create () in
    fork (fun () -> filter reader);
    for i = 2 to 1000
    do
      writer i;
      yield ()
    done;
    writer (-1);
    exit ()
  ) in
GreenThreads.scheduler main;;

```