Writing Applications using Contiki

Norman Vetter

30. Juli 2015

Structure

Structure

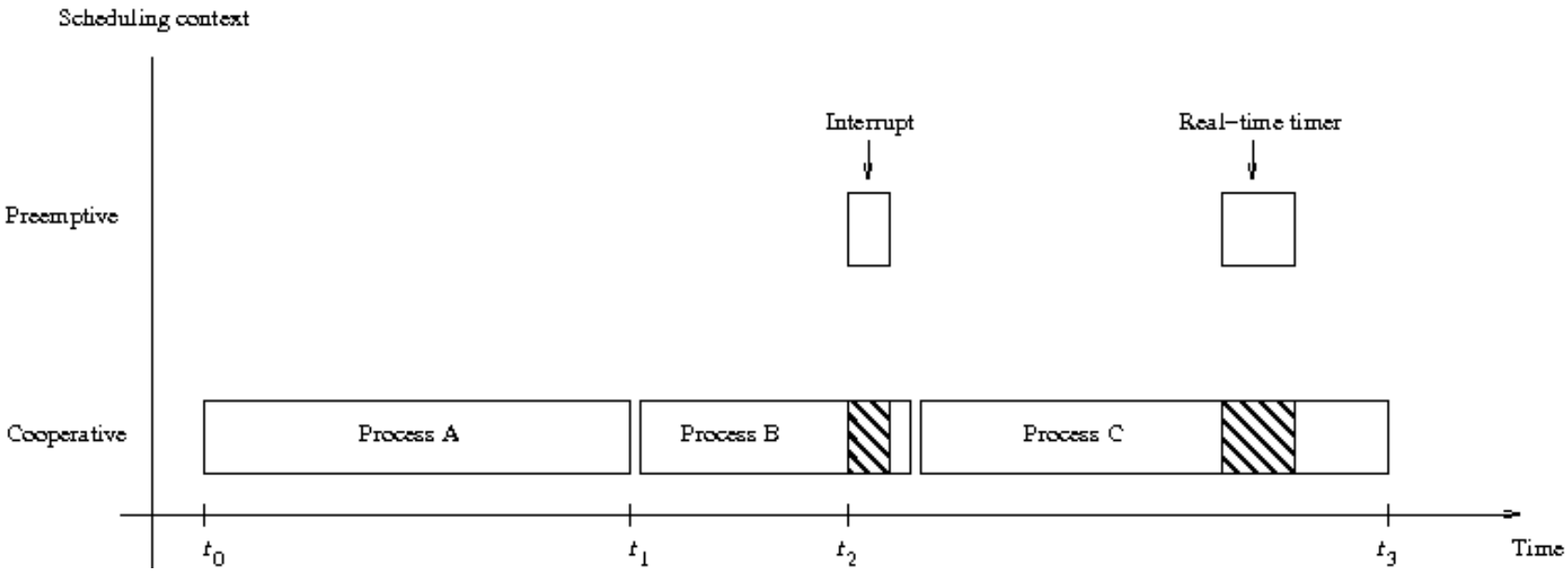
Features

- Open Source Software
- Community and Commercial Support
- IP Networking
  - IPv4, IPv6
  - 6lowpan, RPL, CoAP
- Rime Network Stack
- Sleepy Routers
- Memmory Allocation (memb, mmem, malloc)
- Simulation Environment (Cooja)

Structure

General

- All programs are processes
- Processes are cooperative
- Interrupts and Real-time timer are preemptive



Process Structure

Process Conrol Block:

```
PROCESS(hello_world_process, "Hello world process");
```

Process Thread:

```
PROCESS_THREAD(hello_world_process, ev, data)
{
  PROCESS_BEGIN();

  printf("Hello, world\n");

  PROCESS_END();
}
```

Events

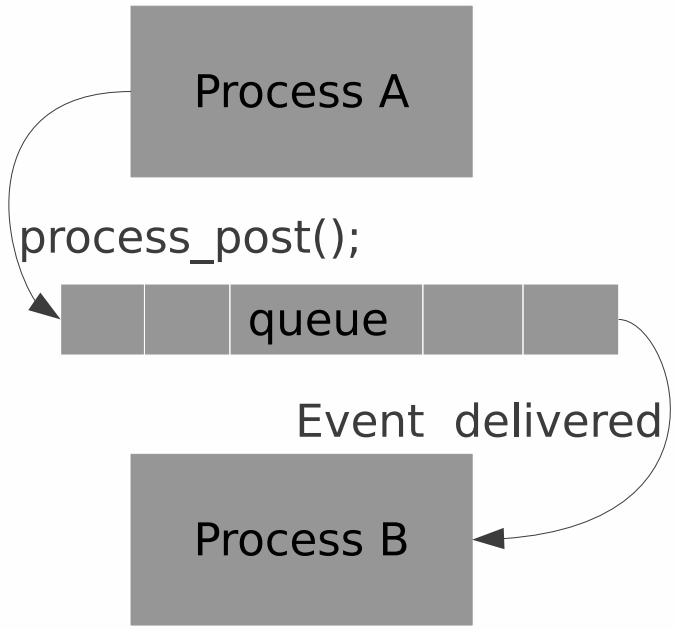


Abbildung: Asynchronous Event.

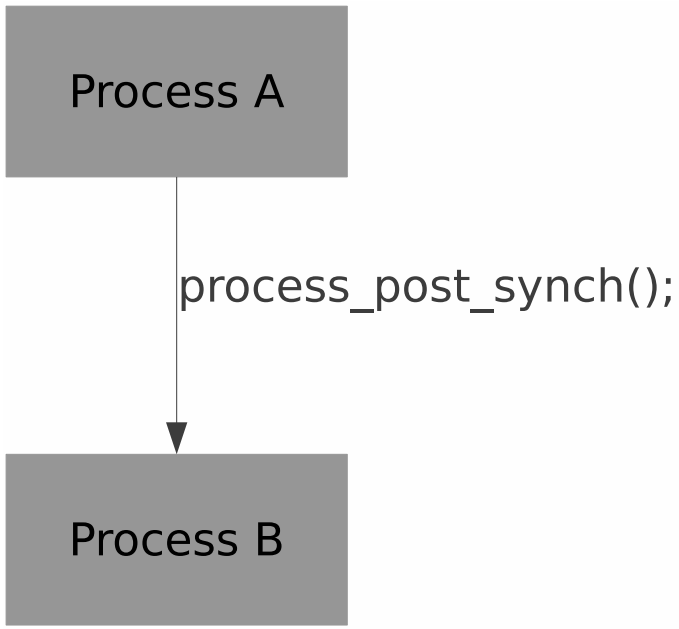


Abbildung: Synchronous Event.

Structure

Clock Module

- handling system time
- block CPU
- base for timer module

Functions:

```
clock_time_t clock_time();                    //Get the system time.
unsigned long clock_seconds();                //Get the system time(s)
void clock_delay(unsigned int delay);//Delay the CPU.
void clock_wait(int delay);                   //Delay the CPU (ticks).
CLOCK_SECOND;                                 //Ticks per second.
```

Timer Module

- Timer: timer, stimer, etimer, ctimer, rtimer
- declared as: struct timer
- timer + stimer has to be checked manually
- etimer throws event: PROCESS_EVENT_TIMER
- ctimer calls a function with data pointer as argument
- rtimer
  - preemptiv
  - uses own clock module with higher resolution