

Orijinal Contiki-NG'de Processlerin Koşturulma Sırası Nasıl Belirlenir ?

Orijinal Contiki-NG'de process'lerin koşturulma sırası FIFO kuyruğuna dayanır. `process_run()` methodu çağrıldığında ilk olarak `do_poll()` methodu ile needspoll flagi true olan processler `process_list` üzerinde sırayla işlenir ardından `do_event()` methodu çağrılarak event kuyruğundaki ilk event alınır ve kuyruk dairesel buffer mantığında ilerletilir. Eğer event broadcast ise `process_list` üzerindeki tüm processlere sırayla `call_process()` methodu ile gönderilir, event broadcast değilse sadece hedef processe iletilir. Processler bağlı listede tutulur ve yeni processler `process_start()` ile listenin başına eklenir. Sistemde herhangi bir priority mekanizması bulunmaz, tamamen event sırası ve process listesindeki konum belirleyicidir.

Yapıların İzahı

1. process.c

- `string.h` include edildi. (process ismine göre luck değeri ataması için)
- `set_luck_by_name()` methodu eklendi bu method belirlenen isimdeki processlere belirlediğimiz luck değeri atamasını yapıyor.
- `find_highest_priority_process()` methodu eklendi bu method process listesindeki önceliği en büyük olan (luck değeri en küçük olan) processi bulur.
- `process_start()` methodu içerisinde `set_luck_by_name()` methodu çalıştırıldı ki processler başlatıldığında istediğimiz processin luck değerini değiştirebilelim.
- `call_process()` methodu içerisinde log kısmında processin luck değerinin de bastırılması sağlandı.
- `do_poll()` methodunun poll edilmesi gereken tüm processleri priority sırasına göre çalıştırması sağlandı.
- `do_event()` methodunun broadcast eventleri en düşük luck değerinden en yüksek luck değerine kadar dağıtması sağlandı.

2. process.h

- Oluşturulacak processe luck değeri eklendi. (başlangıçta her process için 64)
- Process struct yapısına luck değeri eklendi.

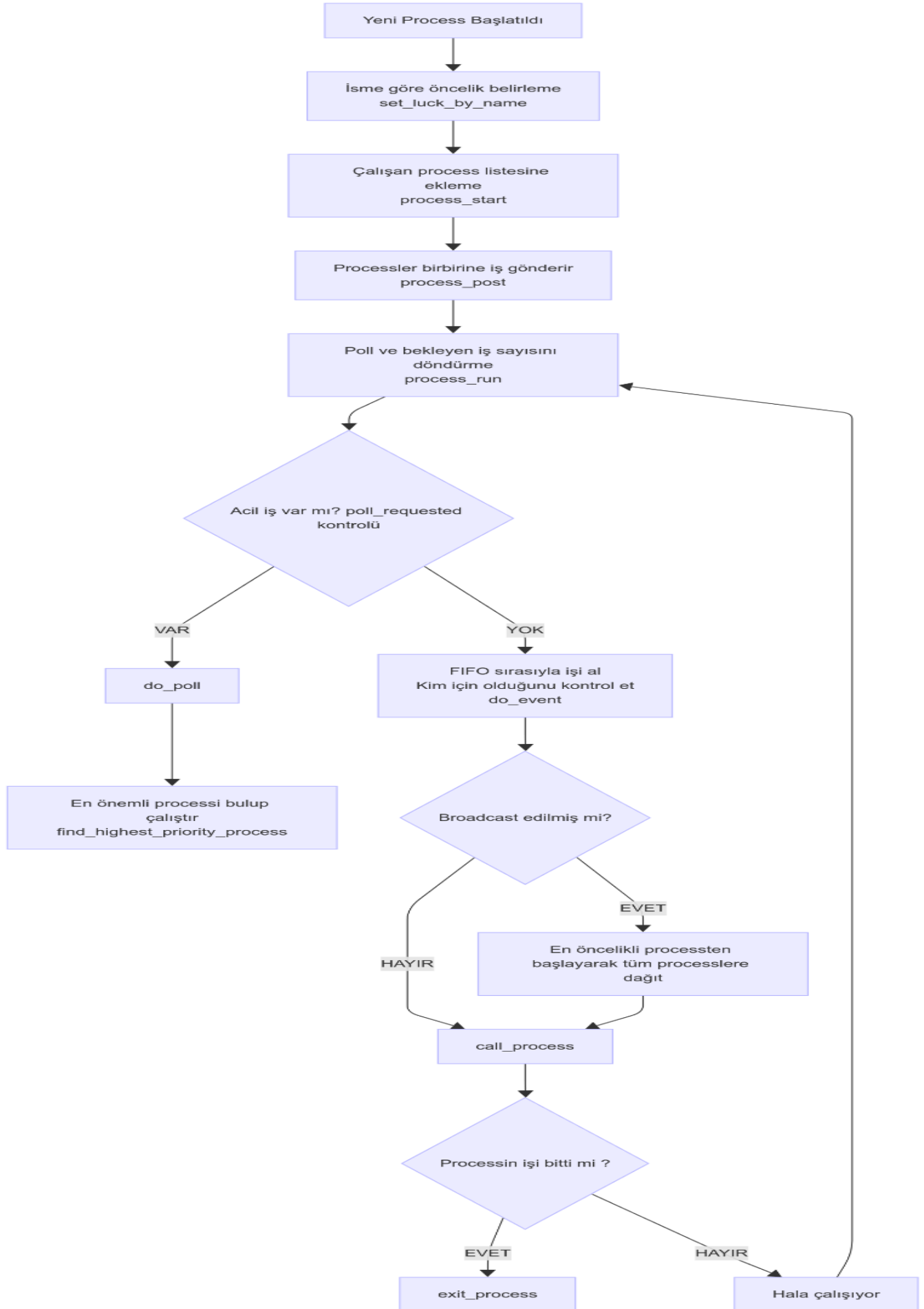
3. hardworker.c

- `printf` ile yazdırılan loglar `LOG_DBG` ile yazdırıldı.

Priority Scheduling ile Çalışan Contiki-NG Nasıl Çalışır ?

Priority Scheduling ile çalışan Contiki-NG'de processlerin koşturulma sırası luck değerlerine dayanır. `process_run()` methodu çağrıldığında ilk olarak `do_poll()` methodu ile needspoll flagi true olan processler arasından `find_highest_priority_process()` methodu ile en yüksek öncelikli process bulunur ve bekleyen tüm processler priority sırasına göre işlenir. Ardından `do_event()` methodu çağrılarak event kuyruğundaki ilk event alınır ve kuyruk dairesel buffer mantığında ilerletilir. Eğer event broadcast ise `process_list` üzerindeki tüm processler öncelik değerlerine göre `call_process()` methodu ile çağrılır. Event broadcast değilse sadece hedef processe iletilir. Processler bağlı listede tutulur ve yeni processler `process_start()` ile listenin başına eklenir ve proje kapsamında belirlenen processler için `set_luck_by_name()` fonksiyonu ile öncelik ataması yapılır.

Proje Akış Şeması



Öncelik Sıralaması

1. Event timer (luck = 1)
2. LED Toggle (luck = 16)
3. Dummy Printer (luck = 32)
4. CC2420 driver (luck = 48)

Process_list Düzeni

Processler process_list üzerinde luck değerlerine göre sıralanmamış biçimde tutulmaktadır. Bu yaklaşım da tercih edilebilirdi fakat process sayısının az olduğu varsayılarak her işlemde luck değeri en küçük processin bulunarak kuyruktan alınması tercih edilmiştir.

NOT: Proje şeması classroomda da paylaşılmıştır şema okunmuyorsa bakabilirsiniz.

Değişiklik yapılan ve sisteme (22060388.zip) yüklenen kaynak dosyaları:

- /contiki-ng/os/sys/process.c
- /contiki-ng/os/sys/process.h
- /contiki-ng/examples/os/hardworker.c

Video linki (erişilebilir) : <https://youtu.be/mF-13Ifp7CE>
