

Beágyazott operációs rendszerek – kidolgozott ellenőrző kérdések

Beágyas kérdések:

1. Mi az a három alapvető állapot, amelyben a taszkok tartózkodhatnak? Továbbá milyen átmenetek lehetségesek (preemptív ütemező esetén)?

- Alapvető állapotok: fut, futásra kész, várakozik.
- Átmenetek: várakozik → futásra kész (bekövetkezett a várt esemény), futásra kész → fut (ütemezési döntés), fut → várakozik (elkezd várni az eseményre), fut → futásra kész (preempció, csak preemptív ütemező esetén).

2. Ismertesse a végtelen ciklusú taszk felépítését!

Egy opcionális inicializációs szakasz után egy végtelen ciklus indul, amiből a taszkot a ciklusban elhelyezett OS hívásokkal lehet várakozó állapotba juttatni annak érdekében, hogy más taszkok is vezérlést kaphassanak.

3. Ismertesse a single-shot taszk felépítést!

A taszk elindul, fut amíg be nem fejezi feladatát, majd törli magát. Törlés előtt esetleg kiválthat olyan eseményeket, melyek más taszkokat indítanak el.

4. Ha az egyes taszkok rendelkeznek közös memóriával, könnyen kommunikálhatnak egymással. Mi a hátránya a közös memóriának (ált. esetben közös erőforrásnak)?

A taszkok nem mindig kezelik atomikus módon a közös memóriát (erőforrást) és egymás futását megszakítva, a memóriát (erőforrást) módosítva a memória (erőforrás) inkonzisztenssé válhat.

Pl.: Ha egy megszakítás időnként egy 16 biten ábrázolt számot frissít, egy másik pedig a számot olvassa, 8 bites platform esetén a megszakításnak köszönhetően a szám egyik fele a régi, a másik az új adatot tartalmazza.

5. Két taszk globális változón keresztül kommunikál egymással. Hogyan küszöbölhető ki ebben az esetben a közös erőforrások problémája?

Kölcsönös kizárással az adott erőforrásra. Ezekre módszerek:

- megszakítások tiltása / engedélyezése
- az ütemező tiltása / engedélyezése
- lock-bit használata (test-and-set)
- semaforok használata

6. Mi a semafor, és mit kell tudni a működéséről?

Egy olyan bit/számláló, ami a kölcsönös kizárást hívatott megoldani. Két alapvető művelete van: P(), V(). A P() a TAS, vagyis a lockolás műveletét valósítja meg, ami a semafor értékét, ha nem 0, 1-gyel csökkenti. Ekkor az adott taszk foglalja az erőforrást. A V() a szabadba állítás művelete, mellyel a semafor értékét eggyel növeljük, ha nem a maximális értéken áll. Ha a maximális érték 1, bináris, más esetben számláló semaforról beszélünk.

7. Az OS időkezelés céljából egy periodikus időzítőt használ (heartbeat timer). Az időzítő két jelzése közt eltelt időt nevezzük óraütésnek. Mekkora pontosság érhető el az OS időkezelő szolgáltatásai által?

1-100 ms pontosság.

FreeRTOS:

8. Hogyan konfigurálható a FreeRTOS?

A Startup kóddal, illetve a Simplicity Studio Configurator menüpontjával.

9. Mikor indul el a FreeRTOS ütemezője?

Az OS-sel együtt fordított alkalmazás indulása és az inicializációs szakasz után.

10. Mik a FreeRTOS taszkok (szálak)?

Az OS futtatási egységei. Nincsenek egymástól védelemmel ellátva, így látják egymás memóriaterületeit, ezzel kommunikálva egymással.

11. Hogyan tudnak egymással kommunikálni a FreeRTOS taszkok?

Közös memórián keresztül.