

Párhuzamos és eseményvezérelt programozás beágyazott rendszerekben szóbeli tételsor

1. Az operációs rendszer és a HW kapcsolata. OS alkalmazásának előnyei és hátrányai. HW hatása a párhuzamos végrehajtásra, HW architektúrák, cache koherencia, spekulatív végrehajtás, megszakítás és fajtái. Valós-idejű végrehajtás és az OS kapcsolata.
2. Védelmi szintek és rendszerhívás, kontextus fogalma. Operációs rendszerek és azok felépítése és OS választás szempontjai. Linux és FreeRTOS főbb eltérései és hasonlóságai.
3. Szekvenciális és párhuzamos programozás. Párhuzamos programozás történe. Tipikus párhuzamos feladatok. Amdahl és Gustafson törvényei és következményeik. Eseményvezéreltség.
4. Feladat fogalma. Feladat egyszerűsített memóriaképe, user és kernel területen lévő információk, egyszerűsített állapot-átmeneti ábra. Linux és FreeRTOS állapot-átmeneti ábrájának ismertetése.
5. Feladatok ütemezése. Ütemezés időskálái. Nem preemptív és preemptív ütemezés és összehasonlításuk. Ütemező választás. FIFO, RR ütemezők és azok összehasonlítása. Prioritásos ütemezők, egyszerű és többszintű sorok ütemezők és tulajdonságaik. FreeRTOS ütemezője.
6. Feladatok ütemezése általában. Többprocesszoros ütemezés és annak kapcsolata az egyprocesszoros esettel, processzor affinitás, terhelés elosztás. Linux ütemezője.
7. Feladatok leírása. Alacsony szintű vezérlési szerkezetek, coroutine és fiber. Folyamatok és szálak részletes bemutatása. Folyamatok és szálak megvalósításának HW feltételei.
8. Folyamatok és szálak FreeRTOS-ban és Linux-ban. Linux folyamat és szál létrehozása, FreeRTOS task (szál) és használata.
9. Feladatok együttműködése általában, erőforrás, közös erőforrás, kritikus szakasz, atomi művelet, újrahívhatóság és kölcsönös kizárás definíciója és azok kapcsolata. RAM és PRAM modell, memória mint közös erőforrás. HTM alapötlete és összehasonlítása a kölcsönös kizárással.
10. Közös erőforrásra várakozás (lock) lehetőségei és annak illeszkedése a feladatok egyszerűsített állapot-átmeneti ábrájához. Lockolás részletessége. Aktív és passzív várakozás, összehasonlításuk. Megadott időre történő várakozás (timeout) lehetőségei és időkezelés.
11. Kölcsönös kizárás lehetőségei közös memória esetén (PRAM modell). Lockbit-től az RCU-ig, a megoldások és azok összehasonlítása. FreeRTOS és Linux példák. Randevú és kommunikáció PRAM modell esetén.
12. Párhuzamos programozás során elkövetett tipikus hibák és azok elleni védekezés. Holtpont, szükséges feltételek, holtpont kezelése. Prioritás inverzió és elkerülése. Monitor és megvalósítása.
13. FreeRTOS és Linux megoldások kölcsönös kizárásra, szinkronizációra közös memória esetén. A megoldások felsorolása, főbb jellemzőik, összehasonlításuk.

14. Folyamatok közötti kommunikáció (IPC) és s közös memória alapú kommunikáció összehasonlítása. Üzenet fogalom és címzés, nyugtázás. Direkt és indirekt, szinkron és asszinkron kommunikáció. IPC megoldások általánosan és FreeRTOS és Linux példák rájuk (részletesen nem kell őket ismertetni).
15. Linux IPC megoldások a jelzésektől a POSIX IPC főbb eleméig (socket-et kivéve).
16. FreeRTOS IPC jellegű megoldásai.
17. Távoli eljárás hívás és annak implementációja Linux alatt.
18. Elosztott rendszer architektúrák. Publish-subscribe, virtuális adatbusz, proxy és bróker részletes ismertetése.
19. TCP/IP protokoll készlet architektúrája és implementációja, tesztprogramjai Linux alatt. Tűzfal alapok. Protokoll analízátorok és működésük alapjai, Wireshark és tcpdump alapok. TCP és UDP alapok, összehasonlításuk.
20. Socket programozás. Socket-ek típusai. Socket használata, tipikus munkamenet TCP és UDP socket esetén.
21. Hibakeresés eszközei és azok használata Linux alatt.
22. Rendszer virtualizáció alkalmazása, hardver és szoftver követelmények. Hypervisor szerepe és alkalmazása beágyazott rendszerekben. Periféria virtualizációs, különös tekintettel a szenzor virtualizációra. (2017-ben ilyen részletességgel nem ment le, nem kérdezzük)