

Käyttöjärjestelmien ja niitä tukevien suorittimien perusrakenne

1. Esimerkkiarkkitehtuurissamme on vain kaksi yleisrekisteriä (R0 ja R1). Mikä on mielestäsi pienin järkevä rekisterien lukumäärä? Mitkä seikat rajoittavat rekisterien lukumäärää?
2. Kerro (esimerkiksi kuvan avulla), mitä tapahtuu, kun jokin seuraavista keskeytyksistä tulee ohjelman suorituksen ollessa käyttäjän moodissa (USER_MODE = true). Milloin kukin keskeytys voi tulla ja mihin käyttöjärjestelmät tyypillisesti näitä keskeytyksiä käyttävät?
 - a. Ohjelmoitu keskeytys (SVC).
 - b. Ohjelmavirhekeskeytys (trap).
 - c. I/O-keskeytys.
3. Jos prosessorin muistiavaruus on lineaarinen, ts. minkäänlaista segmentointia (segmentillä on base+limit-suojaus) ei ole käytössä, ei useamman ohjelman samanaikainen muistiin lataaminen ole aivan ongelmatonta. Hahmottele ratkaisu, jonka avulla ohjelma voidaan ladata mielivaltaiseen paikkaan muistia ja ajaa sitä siellä. Millä tavoilla prosessoriarkkitehtuuri voi helpottaa mielivaltaisessa paikassa ajettavan koodin kirjoittamista?
4. Minkä seuraavista toimenpiteistä pitää ehdottomasti olla kiellettyjä käyttäjän moodissa (USER_MODE = true) [Kirjan tehtävä 1.6]:
 - a. Lue käyttöjärjestelmän alueelta (LOAD).
 - b. Aseta TIMER-rekisteri.
 - c. Estä keskeytykset (IOF).
 - d. Salli keskeytykset (ION).
 - e. Hyppää käyttöjärjestelmän alueella olevaan osoitteeseen.
 - f. Käsky, joka pysäyttää suorittimen seuraavan keskeytyksen tapahtumiseen asti (WAIT).
5. Moniajokäyttöjärjestelmiä on tehty myös alkuaikojen mikroprosessoreille, joissa ei ollut muistinsuojauksesta tietoaakaan. Millaisia vaatimuksia tällainen järjestelmä asettaa siinä ajettaville ohjelmille? Onko järjestelmä mahdollista suunnitella siten, että ohjelmat varmasti ovat suojattuja toisiltaan? Mitä porsaanreikiä järjestelmään saattaa jäädä? [Muunneltu kirjan tehtävä 1.7]