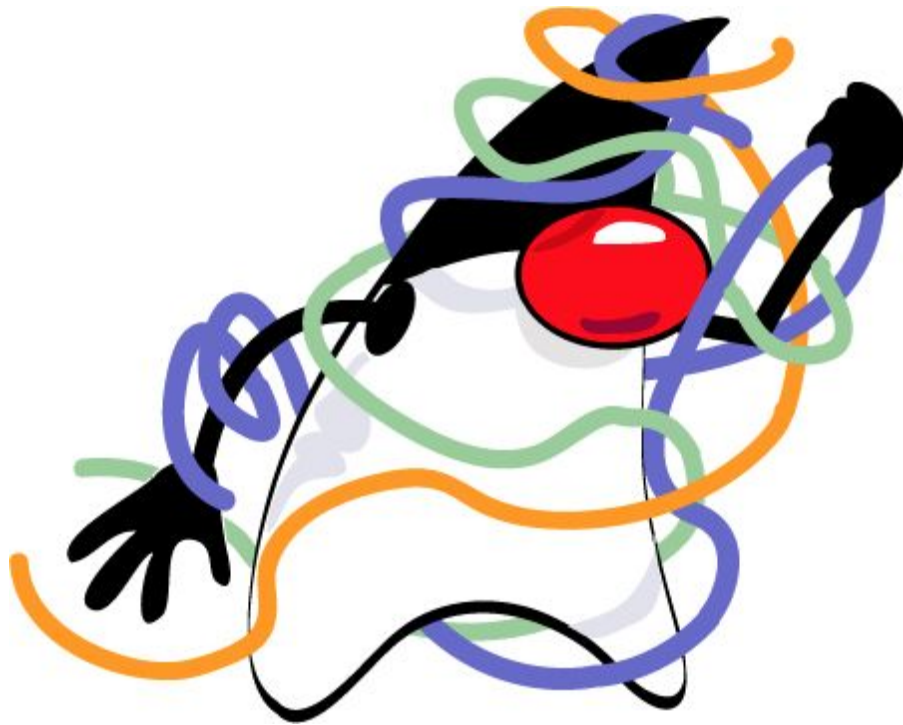


Programazio Konkurrentea



Alain Barrero
2016-2017

Konkurrentzia. Sarrera

Programazio konkurrentea hainbat ataza interaktibo aldi berean exekutatzean datza. Horregatik, desberdintasunak sortzen dira programazio sekuentzialaren eta konkurrentearen artean, prozesuen kantitatean adibidez, sekuentzialean prozesu bakar bat erabiltzen dugu, kontrruntzian ordea, hainbat prozesu. Ondorioz, programazio konkurrentean aplikazioaren errendimendua eta erantzun denbora haunditzen du.

Prozesuak eta Hariak

Prozesuak ekintza atomikoen sekuentziak dira, hauek egoera finituko makinekin modelatzen dira, etiketaturiko trantsizio-sisteman ereduak sortzeko, ondoren javaz programatuak izateko, javako hariekin.

Garrantzitsua da agindu eta portaera guztiak ongi menderatzea, programa konkurrenteak ahalik eta errexen sortzeko.

Exekuzio Konkurrentea

Programa konkurrenteak modelatuak izateko konposizio paraleloa (\parallel) erabiltzen dugu, bi prozesu deberdin batzeko.

Konposizioan interakzioa modelatzen da, hau da, konpartituriko ekintzak eta ez konpartituriko ekintzak hartzen dira kontutan. Konpartitua ez den ekintza bat modu arbitrarioan izan daiteke tartekatua, konpartituriko ekintza ordea, konpartitzen duten prozesu guztiek aldiberean exekutatuko dute. Ez dira erabiltzen konpartituriko ekintzak, konpartituriko baliabide bat adieraziko duen beste prozesu pasibo bat baizik.

Objektu Konpartituak eta Elkar Bazterketa

Java metodoen aktibazioak ez dira atomikoak, hau da, objektu konpartitu bat aldi berean egikaritu dezakete. Horrelako egoerak ekiditeko elkar bazterketa erabiltzen da, metodoan **synchronized** hitz erreserbatua erabiliz.

Monitoreak eta baldintzen sinkronizazioa

Programazio konkurrentean, entitate aktiboak (ekintzak hasten dituztenak) hariekin imlementatzen dira eta entitate pasiboak (ekintzei erantzuten dietenak) monitorekin imlementatzen dira.

Elkar Blokeaketa

Sistema elkar-blokeaketa hematen denean, ezin dugu egin aurrerapenik. Hori dela eta, programazio konkurrentean horrelako egoerak saihesten saiatzen dira.

Horrelako egoerak ekiditeko, zenbait arau hartu behar ditugu kontutan. Elkar-bazterketa zihurtatzea, eskuratze inkrementala, prozesuek solik borondatez askatu ditzatela baliabide konpartitua eta zain ziklo bat ez dela ematen ziurtatzen badugu.

Segurtasun eta Bizitasun Propietateak

Segurtasun-propietate batek adierazten du ez dela ezer txarrik gertatuko eta bizitasun propietate batek adierazten du zerbait ona noizbait gertatuko dela. Horretarako, aurrerapen propietateak erabiltzen ditugu. Bizitasun propietateak adierazten du beti emango dela kasua non ekintza bat noizbait izango den egikaritua.

Denbora Errealeko Sistemak

Denbora Errealeko Sistema (DES) bat bere prozesuen funtzio guztiak emandako denbora murriztapenen barruan exekutatzen dituen sistema kontrolatu bat da. Sistemaren erantzuna zuzena izango da: emaitza logikoak onak badira eta epean ematen badira.

Sistemaren erantzuna zuzena den konprbatzeko lehentasunetan oinarritutako planifikazioa erabiltzen dira. Rate Monotonic (RM) periodo txikienekoari lehentasun handiagoa ematen dio, Earliest Deadline First (EDF) epe hurbilenekoari lehentasun handiagoa ematen dio eta plangintza puntuen teorema, ataza bakoitza bere lehen periodo bukaera baino lehenago exekutatu daitekeen egiaztatezen du, plangintza puntuak aztertuz.