

Miniprojekt: Scheduler

Cél:

Egy Round-Robin elven működő és szignálokat alkalmazó feladatütemező létrehozása.

Feladat:

Írj egy C nyelvű programot, amely a feladatokat és az ütemezőt implementálja! Az ütemezendő feladatokat eljárás segítségével valósítsd meg! Ezek futásideje véletlenszerűen néhány másodperce legyen (pl. <20s)!

A program fő programegysége (`main`) legyen az ütemező! Ez ciklikus működése során időnként (véletlenszerűen) hozzon létre egy gyerek folyamatot, ami majd az adott feladatot futtatja. Az ütemező persze tartsa nyilván a létrehozott gyermek folyamatokat, azok PID-jét egy pipe segítségével megvalósított „ütemezési sor” (FIFO) segítségével tárolja! (Ide kerüljön be minden újonnan létrehozott gyerek folyamat!) Az ütemezési sorban következő gyermek folyamatnak küldjön az ütemező egy `SIGUSR1` szignált („aktiválás” céljából) és kerüljön várakozó állapotba (`pause`) természetesen csak az után, hogy a megfelelő adminisztrációt elvégezte! Az ütemező `SIGUSR2` vagy `SIGCHLD` szignál érkezése esetén alvó állapotból „újra aktiválódva” folytassa ciklikus működését. Utóbbi szignál esetén könyvelje el az adott feladat, azaz a gyerek folyamat befejeződését, míg előbbi szignál esetén a korábban az ütemezési sor elején szereplő feladatot helyezze a sor végére! Az ütemező `SIGINT` vagy `SIGTERM` szignál érkezése esetén állítsa le az összes létrehozott gyermek folyamatot és ő is álljon le!

Minden gyermek folyamat létrehozása után egyből kerüljön alvó állapotba! A kiválasztott, azaz `SIGUSR1` szignált kapott gyermek folyamat állítson be egy pár másodperces (pl. 4s) időzítőt az `alarm` függvényvel, majd kezdje el vagy folytassa a működését! Ha egy futó gyermek folyamat `SIGALRM` szignált kap (azaz lejár a futásra szánt időrés), akkor küldjön a szülőjének, vagyis az ütemezőnek egy `SIGUSR2` szignált és kerüljön vissza várakozó állapotba! (Minden gyerek folyamat alapértelmezetten küld egy `SIGCHLD` szignált a szülőjének, jelenleg az ütemezőnek, abban az esetben, ha befejeződött. Halott folyamatok ne legyenek újra ütemezve!)