

Operációs rendszerek

3. feladat (IPC)

Alapelvek:

A feladatokat nem specifikáljuk pontosan, inkább csak ötleteket szeretnénk adni, a megvalósítás módját ti választjátok meg. Annyi megkötést teszünk, hogy a processzek közötti kommunikációban csak a szorosabban vett eszközöket alkalmazzátok:

- Message
- Semaphore
- Signal
- Event flag
- Pipe
- Shared memory

Minden feladatnak karakteres terminál szinten kell működnie. Ez nem jelenti azt, hogy nem lehet igényesebb módon kezelni a képernyőt, ismerkedjétek meg a különféle ESCAPE szekvenciákkal, vagy az azokra épülő eljáráskönyvtárakkal, amilyen pl. a „curses.h”-ben található. Így primitív „grafika” is megvalósítható.

Egy-egy feladatban többen is együtt kell, működjenek. Az együttműködést olyan formában képzelem el, hogy az általánosan használható programelemek (pl. felhasználói felület) csoportszinten megcsinálják (ez nyilván egy függvénykönyvtár, amit csak linkelni kell), a feladat-specifikus elemeket pedig egy-egy programba osztva, a kommunikációs szabályok és egyebek egyeztetése után külön-külön készíthetitek.

Feladatok:

1. Gyártócella-szimuláció

Amikor egy gyártócellát telepítenek, vagy üzemeltetnek, nem lehet előre minden esetre felkészülni. Hogy ne adódjon komoly anyagi károkat jelentő fennakadás vagy baleset, és lehetőleg optimálisan működtessük a gépeinket, szimulációkon próbáljuk ki a folyamatokat.

A gyártócella felépítése a következő:

Nyersanyag/készáru raktár (A)
Robotkocsi (K)
Előmunkáló gép (EG)
Robot (R)
Finommegmunkáló gép (FG)

Minden egyes egységnek egy-egy program felelne meg. A feladatok elosztásának egy lehetséges módja:

Legyen minden munkadarab útja ugyanaz: A – K – EG – R – FG – K – A

A raktár tehát külső megrendelésből tölti a készletet, kiszolgálja a belső igényeket, eltárolja a készterméket és kiadja a szállítóknak.

A robotkocsi feladata a nyers munkadarabok szállítása EG részére, valamint a kész munkadarabok raktárra vitele.

Az első megmunkálógép rendelkezik egy ún. palettával, melyen a megmunkálásra váró darabokat tárolja, valamint egyszerre csak egy munkadarabon dolgozhat.

A robot feladata a két megmunkálógép közötti szállítás.

A második gép szintén csak egy darabon tud dolgozni, az elkészült alkatrészeket a saját palettájára teszi.

PI:

A raktár észleli, hogy készlete a biztonsági tartalék alá esett. Gyorsan rendelést ad ki a beszállítónak – itt egyszerűen generál különféle munkadarabokat. A robotkocsi egy darabért fordul a aktárhoz, ahonnan azt meg is kapja, majd viszi az első géphez. Ott azonban tele van a paletta, ezért várnia kell, amíg nem tudja letenni. Csakhogy közben a másik gép befejezett egy alkatrészt, tenné a saját palettájára, de az tele van. Mivel a kocsi nem tudja elvinni, kialakul egy körbevárakozás, amit dead-locknak is neveznek. Az ilyen helyzeteket ki kell küszöbölni.

2. Benzinkút

Adott egy benzinkút az adott városba vezető 3 főútvonal találkozásánál. Az egyes útvonalaknak ismertek az átlagos fogalmi adatai az egyes napszakokban. Az autóknek bizonyos százaléka megáll a benzinkútnál (de ez is véletlenszerű... tehát semmiképpen nem úgy, hogy minden 10. autó megáll... lehet hogy éppen 2 egymás utáni áll meg... stb) és valamilyen mennyiségű (véletlenszerű) benzint tankol (ami legalább a tank méretének fele). 1 kútfej van, és 1 liter tankolásának ideje ismert. A benzinkút benzintartályának mérete ismert, amennyiben egy bizonyos jelzőszint alá megy a készlet, rendelni kell! A kiszolgálás ideje szintén ismert. Amennyiben a benzinkútnál legalább 10 kocsi várakozik, a kocsik megállásának valószínűsége az eredeti tizedére esik. Amennyiben a forgalom (időegység alatt eladott benzin mennyisége) egy meghatározott időintervallumban egy bizonyos szint alá esik, akkor a benzinkút akciót hirdet egy időszakra és ilyenkor a benzinkútra való betérés valószínűsége 100%-al megnő. Készítse el a feladat szimulációját.

3. Olaszkonyha

Adott öt olasz, akik csak esznek és beszélnek. Egy kerek asztal körül élnek, mindegyik előtt egy-egy tál spagetti – amihez két villa kell. A probléma az, hogy minden két tál között csak egy villa van, azt kell felváltva használniuk. Hogyan lehet megvalósítani, hogy mindannyian egyenek is és jusson idő a beszélgetésre is? Vigyázzon: aki keveset eszik, de sokat beszél az éhen marad, aki viszont sokat eszik, de keveset beszél az pedig elhízik! (Próbálja úgy megoldani a feladatot, hogy akár 6, 7, 8 résztvevő estén is működjön a dolog.)

4. Színdarab

A rendező választ egy színdarabot, meghatározza a szereposztást és levezényli az előadást.

IPC módon a rendező egy processz, elolvassa a szereposztást, annyi színészt hív – azaz processzt hoz létre – ahány szereplő van. Kiosztja a szövegekönyveket a színészeknek, és minden színészt figyelmeztet, amikor az ő szövege következik (az előadás során a színészek követik az előadást). A színészek tévedhetnek is (vagyis rossz szöveget mondhatnak), ilyenkor a közönség felhőrdül, és állapota gondterheltebbre változik (-1 az állapotukban), viszont sok jó mondat után egyre jobban élvezik az előadást (állapotuk +1-el változik). Ha az állapotuk egy bizonyos szint alá megy, egyszerűen kivonulnak a színházból és hazamennek, ha pedig jó volt az előadás, akkor tapsolnak.

(Az Interneten könnyen elérhetően egész színdarabok. Jó lenne néhányat végigjátszani.)

5. Repülésirányító

A légtérben n számú repülőgép várakozik leszállásra. A repülőtéren 1 db leszállópálya van. Repülőgépek érkeznek a légtérbe, akik le akarnak szállni, és repülőgépek várakoznak felszállásra a földön. A légtérben tartózkodó repülőgépeket hozza le egy üres leszállópályára, a földön tartózkodókat pedig indítsa útnak az alábbi prioritásokat figyelembe véve: legfontosabb azokat a gépeket lehozni, akiknek az üzemanyagszintjük egy bizonyos szint alá süllyedt (a levegőben tartózkodó gépek egy adott idő alatt ismert mennyiségű üzemanyagot fogyasztanak el), utána ügyelni kell, hogy a gépek minél kevesebbet várjanak mind a felszállásnál, mind a leszállásnál.

A lehozás folyamata a következő:

Üres a leszállópálya – gépkiválasztás – üzenet, hogy elkezdheti az ereszkedést – folyamatos ereszkedés vizsgálata (eltérések esetén figyelmeztetés) – földet-érés sikerességének visszajelzése (sikertelenség esetén leszállópálya lezárása, az elegendő üzemanyaggal rendelkező gépek más repülőtérré való átirányítása) – leszállópálya elhagyásának visszajelzése leszállópálya felszabadítása (jöhet a következő). (felszállás esetén hasonló logika szerint kell eljárni)

6. Repülőjegy-foglaló rendszer

N db utazási irodában lehet m db repülőjáratra helyet foglalni. (az utazási irodák külön processzek) Az egyes járatoknál ismert a maximális befogadóképesség. Úgy oldja meg a foglalást úgy, hogy ne történhessen meg az, hogy 2 utasnak is ugyanarra a helyre foglalnak helyet 2 különböző utazási irodában. A helyfoglalást le is lehet mondani!

7. Párbeszéd

Hasonló a talk programhoz, de mindig csak az egyik fél adhat, a másiknak meg kell várni, amíg megkapja a szólás jogát – vezérelés (token) protokoll.

8. Fórum

Folyik n db beszélgetés, amibe bárki bekapcsolódhat. A résztvevők beceneven azonosíthatóak, egyszerre egy adott beszélgetésben csak egy beszélhet, de mindenki láthatja a teljes beszélgetést. A válaszolók sorban állnak a válaszáért, vagy szólásért különböző prioritások alapján. A prioritásokat a moderátor szabja meg, aki a beszélgetést felügyeli. Aki nem odaillo válaszokat, hozzászólásokat mond, annak a moderátor csökkenti a prioritását (egy bizonyos szint alatt ki is zárhatja a beszélgetésből), aki jókat mond, annak növeli (szerencsére az internetes fórumok ennél szabadabbak... csak a feladat miatt vezettük be a cenzúra ezen durva változatát, vagyis ezt senki ne tartsa irányadónak a fórumok rendjével kapcsolatosan!!!)

9. A majmok etetése tilos! ... Vagy mégsem?

Az alábbi feladatot kell megoldanunk. Egy szigeten 2 majomcsapat él, akiket egy folyó választ el egymástól. Az egyik csapat a folyó egyik oldalán él, a másik a folyó másik oldalán él. A folyó felett két kötél van kifeszítve, amelyek egy-egy adott terhelést bírnak ki. Nem is lenne semmi baj, csak hogy a bal oldali parton élő majmok a jobb oldali parton termő banánt szeretik, a jobb oldali parton élők pedig a bal oldali parton termőt! Úgyhogy rendszeresen átjárnak kötélen függeszkedve az egyik partról a másikra... Oldja meg, hogy ne induljanak el egy időben a két partról egymással szemben egy kötélre! (Vagyis egy kötélre egy időben csak egyirányú fogalom legyen.) A majmok súlya ismert, de amikor már teleették magukat banánnal a megfelelő parton, akkor természetesen nagyobb, mint éhgyomorral. Minden majomnál egy adott intervallumban lévő számmal jellemezhető hogy mennyire éhes (pl.: 0-100, ahol 100 jelenti, hogy egyáltalán nem éhes, a 0 pedig hogy éhen halt.). A majmok az idő múlásával egyre éhesebbek lesznek, de különböző sebességgel. Oldja meg, hogy egy majom se haljon éhen és azt is hogy ne legyenek távol a csapatuktól túlságosan sokáig!

10. Étterem

Ön egy étterem tulajdonosa. Az alábbi módon jár el: A vendégek az étterem előtt várakoznak bebocsátásra (oda érkeznek különböző időközönként). Amennyiben van szabad hely valamelyik asztalnál, kimegy egyért és leülteti az adott helyre, felveszi a rendelést és rendelkezik a kiszolgálásról. Közben figyeli, hogy melyik vendég ül már legalább fél órája az étteremben, és ha talál ilyet, akkor megkérdezi, hogy akar-e még rendelni és ha nem, akkor kiküldi az étteremből... (A vendégek maguktól is felállhatnak és elmehetnek az étteremből a nélkül, hogy fel kellene szólítani őket!) Oldja meg az adott feladatot IPC mechanizmusok segítségével!

11. Éttermi kiszolgálás

Egy étterembe vendégek érkeznek véletlenszerű időközönként. Vegye fel a rendelést és juttassa el a konyhába. A konyhában állítsa be a rendeléseket egy sorba. Vannak visszatérő vendégek, akiknek a rendelése magasabb prioritású. Időnként a konyhában az ételek odaégnak, ilyenkor az egész ételt újra meg kell csinálni... A kész ételeket ki kell vinni a vendégnek és figyelni kell, hogy a vendég visszatérő vendég vagy sem... Visszatérő vendégek esetén a prioritást az határozza meg, hogy hányadik alkalommal tértek be az adott étterembe (de lehet az is hogy összesen eddig mennyi pénzt költöttek...).

12. Sakk játék

Az ismert játék megvalósítása. Figyelni kell, hogy felváltva lehet csak lépni, és azt is, hogy az adott lépés lehetséges-e. (Lehet-e oda lépni, és nincs-e éppen sakk, vagy a lépéstől nem lesz e sakk állás stb.), és hogy matt állás van-e éppen. (vagyis vége a játéknak)

13. Dominó játék

Az ismert játék megvalósítása. (Hasonlóan a sakkhöz.) No comment.

14. Dáma játék

Az ismert játék megvalósítása. (Hasonlóan a sakkhöz.) No comment.

Megjegyzés

A feladatokra csak a megadott számú hallgató jelentkezhet! A feladat forráskódját a gyakorlatvezetőnek kell elküldeni a@iit.uni-miskolc.hu címre, majd személyesen kell a csoportnak ismertetni a megoldás menetét!

A feladat beadásának határideje: a szorgalmi időszak utolsó hete.

Jó munkát! Jó szórakozást!