

	Feladat
	1. Írjon egy C programot, amely egy szülőprocessz révén készít egy gyermekprocesszt, a gyermekben futtasson egy másik programot az execl() hívással(Environemnten keresztül kapja meg, hogy mit indítson el a program), mely kiírja a PID-jét és szülője PID-jét, majd a szülő is kiírja mi a PID-je és a gyermeke PID-je.
	2. Írjon egy C programot, amely létrehoz egy osztott memória szegmenst és majd rácsatlakozik. Továbbá egy másik program olvasson be 3 számot egy file-ból(amik a háromszög oldalainak hosszát jelentik) az osztott memóriába és döntse el, hogy szerkeszthető-e belőlük háromszög. A döntési eredmény a file kimeneten, ha készíthető háromszög van kerülete illetve területe, ha nincs akkor ezekre -1-et ad vissza. Az adatokat és az eredményt egy fájl kimeneten adjuk vissza.
	3. Írjon C nyelvű programot, ami: létrehoz két gyermekprocesszt ezek a gyermekprocesszek létrehoznak 3-3 további gyereket ezek az unokák várakoznak néhány másodpercet és szünjenek meg a szülők várják meg a gyerekek befejeződését és csak utána szünjenek meg.
	4. Írjon C nyelvű programot, ami létrehoz két csövezeteket (két file deskriptor part) elforkol a szülő elküldi a saját pid-jét a gyermeknek az egyik csövön a gyermek kiírja a képernyőre és visszküldi egy az egyet a másik csövön megszűnnek a processzek (a szülő megvárja a gyereket)
	5. Írjon C nyelvű programot, ami létrehoz egy csövezeteket (egy file deskriptor part) elforkol a gyermek egy signal hatására elküld egy rövid szöveget a szülőnek (a signalig blokkol) a szülő kiírja a képernyőre megszűnnek a processzek (a szülő megvárja a gyereket)
	6. Írjon C nyelvű programokat, ami létrehoz egy nevesített csövezeteket (bejegyzés az fs-en) megnyitja olvasni próbál belőle közben egy másik program ír bele egy random számot az első program kiírja a kapott számot és visszküldi ennek a dupláját a másik program kiolvassa és kiírja a képernyőre a processzek megszűnnek és a második program eltűnteti a nevesített csövezeteket
	7. Írjon C nyelvű programokat, ami létrehoz egy fájlt az egyik program ír bele és vár pár másodpercet bináris szemafor segítségével "vedi" az írást a másik program pedig kiolvassa belőle a fájl elejétől kezdve

	<p>8. Készítsen egy programot, ami egy 1000 elemű egész szám típusú tömbben úgy keresi meg a maximumot, hogy 10 szálal futtat párhuzamosan, amik közül mindegyik 100 elemet vizsgál meg. Az eredeti szülő processz nem számol, viszont ő gyűjti be az eredményeket, amit pipe-on keresztül vár a processzektol.</p>
	<p>9. Készítsen olyan C programot, amely egy nevesített csővezetéken keresztül fogad szavakat és ezeket egy fájlba menti egy sorszámmal együtt. Illetve egy olyan kliensprogramot, ami ebbe a csőbe adatokat helyez el, amit a felhasználótól kér be. A fő program a SIGTERM szignál hatására szüntesse meg a nevesített csövet és lépjen ki.</p>
	<p>10. Írjon egy olyan C programot, mely egy fájlból számpárokat kiolvassza meghatározza a legnagyobb közös osztóját. A feladat megoldása során használjon message queue(üzenetsoros) IPC mechanizmust, valamint a kimenet kerüljön egy másik fájlba. A kimeneti fájl struktúrája kötött!</p> <p>Példa a bemeneti és kimeneti fájl struktúrájára:</p> <p>Bemeneti fájl: i (Ez jelzi a számpárok darabszámát) x y</p> <p>Kimeneti fájl(Az x,y jelzi a bemeneti adatokat a z pedig a kimenet eredményét): x y z</p>
	<p>11. Írjon egy C programot, ami egy másodfokú egyenlet megoldóképletét reprezentálja nevesített csővezeték segítségével. A műveletvégzéshez szükséges adatokat egy bemeneti fájlból olvassa be, majd az adatokat és az eredményt adja vissza egy kimeneti fájlba. A Bemeneti ill. kimeneti fájl struktúrája kötött!</p> <p>Példa a bemeneti és kimeneti fájl struktúrájára:</p> <p>Bemeneti fájl: i (A bemenő adatok darabszáma) a b c</p> <p>Kimeneti fájl: a b c x y (Az a,b,c jelzi a bemeneti adatokat, az x,y pedig a kimeneti eredményeket)</p>
	<p>12. Írjon C nyelvű programokat, ami hozzon létre egy osztott memória szegmenst ebbe olvassa be egy nagy fájl tartalmát egy másik program pedig olvass ki az osztott mem szegmenből és írja bele egy másik fájlba végül szüntesse meg az shm szegmenst</p>
	<p>13. Írjon C nyelvű programokat, ami létrehoz egy osztott memória szegmenst az egyik program ír bele és vár pár másodpercet bináris szemafor segítségével "vedí" az írást a másik program pedig kiolvass belőle</p>

	<p>14. Írjon C nyelvű programokat, ami hozzon létre egy osztott memória szegmenst a felhasználótól olvasson be szöveget, és ezt írja be az osztott memória területére és küldjön signált a fogadó felnek, hogy kész az üzenet (SIGUSR1) (segítségképpen a másik program pid-je fixen beletelhető a programba) a másik program pedig olvass ki az osztott mem szegmensből, de csak egy adott signal hatására (SIGUSR1) végül szüntesse meg az shm szegmenst</p>
	<p>15. Írjon C nyelvű programokat, ami létrehoz egy üzenetsort SIGHUP signal hatására beletesz egy üzenetet ebbe az üzenetsorba SIGTERM hatására szűn meg az üzenetsor erőforrás és lépjen is ki a program a másik program pedig: SIGHUP signal hatására kiolvasson ebből egy üzenetet, és kiírja a képernyőre</p>
	<p>16. Írjon egy olyan C programot, mely egy fájlból számpárokat kiolvasson meghatározza a legnagyobb közös osztóját. A feladat megoldása során használjon nevesített csővezetékot, valamint a kimenet kerüljön egy másik fájlba. A kimeneti fájl struktúrája kötött!</p> <p>Példa a bemeneti és kimeneti fájl struktúrájára:</p> <p>Bemeneti fájl: i (Ez jelzi a számpárok darabszámát) x y</p> <p>Kimeneti fájl(Az x,y jelzi a bemeneti adatokat a z pedig a kimenet eredményét): x y z</p>
	<p>17. Írjon egy olyan C programot, mely egy fájlból számpárokat kiolvasson meghatározza a legnagyobb közös osztóját. A feladat megoldása során használjon shared memory(osztott memória szegmens) IPC mechanizmust, valamint a kimenet kerüljön egy másik fájlba. A kimeneti fájl struktúrája kötött!</p> <p>Példa a bemeneti és kimeneti fájl struktúrájára:</p> <p>Bemeneti fájl: i (Ez jelzi a számpárok darabszámát) x y</p> <p>Kimeneti fájl(Az x,y jelzi a bemeneti adatokat a z pedig a kimenet eredményét): x y z</p>

	<p>18. Írjon egy olyan C programot, ami egy bemeneti fájlból 3 adatot olvas ki (háromszög oldalai) és eldönti, hogy szerkeszthető e belőlök háromszög. A feladat megoldása során használjon message queue(üzenetsoros mechanizmust), valamint a kimenet kerüljön egy másik fájlba. Ha szerkeszthető belőlük háromszög adjon vissza 1-et, különben pedig 0-t. A ki/bemeneti fájl struktúrája kötött!</p> <p>Példa a bemeneti és kimeneti fájl struktúrájára:</p> <p>Bemeneti fájl: x y z</p> <p>Kimeneti fájl(A q jelzi a visszatérési értéket, tehát hogy szerkeszthető e háromszög): x y z q</p>
	<p>19. Írjon egy olyan C programot, ami egy bemeneti fájlból 3 adatot olvas ki (háromszög oldalai) és eldönti, hogy szerkeszthető e belőlök háromszög. A feladat megoldása során használjon nevesített csővezetékét, valamint a kimenet kerüljön egy másik fájlba. Ha szerkeszthető belőlük háromszög adjon vissza 1-et, különben pedig 0-t. A ki/bemeneti fájl struktúrája kötött!</p> <p>Példa a bemeneti és kimeneti fájl struktúrájára:</p> <p>Bemeneti fájl: x y z</p> <p>Kimeneti fájl(A q jelzi a visszatérési értéket, tehát hogy szerkeszthető e háromszög): x y z q</p>
	<p>20. Írjon egy C programot, ami egy másodfokú egyenlet megoldóképletét reprezentálja osztott memória szegmens segítségével. A műveletvégzéshez szükséges adatokat egy bemeneti fájlból olvassa be, majd az adatokat és az eredményt adja vissza egy kimeneti fájlba. A Bemeneti ill. kimeneti fájl struktúrája kötött!</p> <p>Példa a bemeneti és kimeneti fájl struktúrájára:</p> <p>Bemeneti fájl: i (A megoldani kívánt egyenletek száma) a b c</p> <p>Kimeneti fájl: a b c x y (Az a,b,c jelzi a bemeneti adatokat, az x,y pedig a kimeneti eredményeket)</p>

	<p>21. Írjon egy C programot, ami egy másodfokú egyenlet megoldóképletét reprezentálja message queue(üzenetsoros) IPC mechanizmus segítségével. A műveletvégzéshez szükséges adatokat egy bemeneti fájlból olvassa be, majd az adatokat és az eredményt adja vissza egy kimeneti fájlba. A Bemeneti ill. kimeneti fájl struktúrája kötött!</p> <p>Példa a bemeneti és kimeneti fájl struktúrájára:</p> <p>Bemeneti fájl: i (A megoldani kívánt egyenletek száma) a b c</p> <p>Kimeneti fájl: a b c x y (Az a,b,c jelzi a bemeneti adatokat, az x,y pedig a kimeneti eredményeket)</p>
	<p>22. Írjon C nyelvű programot, ami: létrehoz egy gyermek processzt az újonnan létrejött processzben vegrehajta a "/bin/ls -R /" parancsot a parameteireivel együtt, terjen vissza valamilyen 0-tól különböző értékkel a szülő várja meg amíg befejeződik a processz, és írja ki a visszatérési értéket.</p>
	<p>23. Írjon C nyelvű programot, amely billentyűzetről beker Unix parancsokat és vegrehajtja őket, pontosan úgy, ahogy a shell.</p>
	<p>24. Írjon C nyelvű programot, amely létrehoz egy gyermek processzt, majd küldjön el egy szöveget a szülőtől a gyerekeknek pipe vezetek segítségével. A gyerek írja ki, hogy hány bajtot olvasott, és jelenítse meg az üzenet szövegét.</p>
	<p>25. Írjon egy olyan C programot, amely létrehozza, olvassa, írja és törli az osztott memóriát! A műveletet a parancssoron keresztül adja meg. Amennyiben egy művelet kiadásakor a közös memória nem létezik a program automatikusan hozza létre azt!</p>
	<p>26. Írjon egy olyan C programot, amely létrehoz egy FIFO állományt és tetszőleges két folyamat között működik. A feladat egy olyan kliens-szerver rendszer (két különálló állományból áll!) létrehozása, amelyben a kliens küld egy számot a szervernek, mire a szerver válaszként visszaküldi a szám negyzetét.</p>
	<p>27. Készítsen olyan C programot, amely egy nevesített csövezeteken keresztül fogad szavakat és ezeket egy fájlba menti egy sorszámmal együtt. Illetve egy olyan kliensprogramot, ami ebbe a csöbe adatokat helyez el, amit a felhasználótól kér be. A fő program a SIGTERM szignál hatására szüntesse meg a nevesített csövet és lépjen ki.</p>
	<p>28. Írjon C nyelvű programokat, ami létrehoz egy üzenetsort ebbe az üzenetsorba betesz egy másodpercenként egy változót ennek a változonak a kezdőértéke legyen az aktuális processz azonosítója a változó mindig beírás után növekedjen 2-vel a másik program pedig: kiolvas</p>
	<p>29. Írjon C nyelvű programokat, ami SIGUSR1 szignál érkezéskor növeli egy változó értékét SIGUSR2 szignál hatására pedig kiírja a képernyőre az aktuális értéket. a másik program pedig: futtataskor szignál-t küld az adott processznek</p>