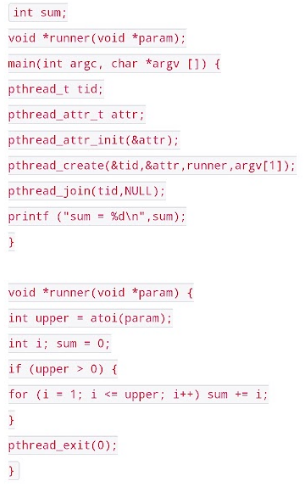
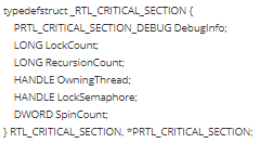
1. Povežite koji od sinkronizacijskih objekata imaju svoj HANDLE, a koji ne.
   1. Kritični odsječak → nema HANDLE
   2. Event (događaj) → ima HANDLE
   3. Semafor → ima HANDLE
   4. Mutex → ima HANDLE
2. Povežite funkcije kod p-niti s time što one zapravo rade.
   1. pthread\_create → Kreira novu nit u stanju ‘spremna’.
   2. pthread\_exit → Završva nit u kojoj se ova funkcija poziva (isto kao i return).
   3. pthread\_mutex\_init → Inicijalizira novi mutex za potrebe sinkronizacije.
   4. pthread\_join → Blokira tekuću nit čekajući da druge niti završe.
3. Povežite osnovne postupke vezane za semafor kao sinkronizacijski objekt s API funkcijama.
   1. Kreiranje HANDLE-a na već postojeći imenovani objekt semafora. → OpenSemaphore
   2. Kreiranje HANDLE-a na novi objekt semafora. → CreateSemaphore
   3. Uvećavanje brojača semafora. → ReleaseSemaphore
   4. Umanjivanje brojača semafora. → WaitForSingleObject
4. Promotrimo višenitni program:
   1. Koji pouzdani višenitni model je korišten u ovom slučaju? → Gazda/šljaker
5. Kojom funkcijom Mailslot klijen odašilje (broadcast) poruku?
   1. WriteFile
6. Povežite ispravno redoslijed povezivanja funkcija za rad s kritičnim odsječkom.
   1. InitializeCriticalSection → 1
   2. EnterCriticalSection → 2
   3. LeaveCriticalSection → 3
   4. DeleteCriticalSection → 4
7. Povežite navedene funkcije kod p-niti s time da li su blokirajuće funkcije ili nisu. (Blokirajuće funkcije općenito blokiraju nit sve dok se nešto ne dogodi nešto što će ih ‘odblokirati’ te nit nastavlj onda dalje s radom).
   1. pthread\_cond\_wait → blokirajuća
   2. pthread\_join → blokirajuća
   3. pthread\_mutex\_lock → blokirajuća
   4. pthread\_mutex\_destoy → nije blokirajuća
   5. pthread\_mutex\_trylock → nije blokirajuća
8. Označite pouzdane višenitne modele za pouzdanu upotrebu:
   1. Server/Klijent
   2. Gazda/šljaker
   3. Model tekuće vrpce
9. Povežite po smjeru komunikacije navedene načine IPC-a:
   1. Socketi → dvosmjerna
   2. Imenovane pipe (cijevi) → dvosmjerna
   3. Anonimne pipe (cijevi) → jednosmjerna
   4. Mailslotovi → jednosmjerna
10. Funkcija CreatePipe koja služi za povezivanje HANDLE-ova kod anonimnih pipa kao prva dva parametra ima adrese od HANDLE-ova koje treba povezati. Uobičajeno je da ta dva HANDLE-a imaju sljedeće pristupe:
    1. Prvi ima GENERIC\_READ, drugi ima GENERIC\_WRITE pristup.
11. Windows API funkcija za kreiranje niti CreateThread ima 6 parametara. Koliko parametara ima istovjetna funkcija kod pniti pthread\_create?
    1. 4
12. P-niti podržavaju slijedeć objekte za sinkronizaciju:
    1. Mutex
    2. Uvjetnu varijablu
13. Na četverojezgrenom procesoru želimo paralelizirati posao obrade N podataka. Koristit ćemo pouzdani višenitni model Gazda/šljaker. Koliko niti će u tom slučaju naš proces (program) ukupno imati?
    1. 5
14. Koliko niti može biti istovremeno u kritičnom odsječku?
    1. 1
15. Promotrimo definiciju strukture za kritični odsječak. Označite tvrdnje koje možemo izvući kao zaključak gledjući tu strukturu.
    1. Kritični odsječci koriste HANDLE od tekuće niti kako bi s njom upravljala.
    2. Ulazak i napuštanje kritičnog odsječka zahtjeva obradu više podataka - nije jednostavno.



1. Označite dobre stvari koje proizlaze iz korištenja višenitnosti:
   1. Dijeljenje resursa (od istog procesa).
   2. Iskorištavanje arhitekture mikroprocesa.
2. Kojom funkcijom se klijent spaja na imenovanu pipu koja je kreirana od serverove strane pomoću CreateNamedPipe?
   1. CreateFile
3. Koliko imamo osnovnih klasa prioriteta procesa kod WIndowsa?
   1. 4
4. CreateThread kod Windows niti vraća kao rezultat HANDLE. Koji tip vraća funkcija pthread\_create kod p-niti?
   1. Int
5. Događaj (event) možemo signalizirati (staviti u signalizirano stanje) pomoću funkcija:
   1. PulseEvent
   2. SetEvent
6. Što se dešava ako kod anonimnih pipa prepunimo pipu šaljući puno više podataka nego što to proces s druge strane može pročitati.
   1. Pisanje s WriteFile u pipu će blokirat sve dok se ne solobodi mjesto u pipi.
7. Upišite znak koji se koristi u UNIX/Linux OS-u za ostvarivanje analognoe stvari kao što su anonimne pipe kod Windows-a, na bazi komandne linije.
   1. |
8. Napisati ime Windows API funkcije kojom se oslobađa vlasništvo niti nad muteksom.
   1. ReleaseMutex
9. Označite načine kako procesi mogu komunicirati ako NISU na istom stroju (računalu) kod Windows-a.
   1. Imenovane pipe (cijevi)
   2. Socket komunikacija
10. Napišite ključnu riječ C-a kojom moramo modificirati deklaraciju globalne varijable kod višenitnog programa.
    1. Volatile
11. Ako želimo koristiti p-niti, ponajviše zbog programske portabilnosti (prenosivosti), onda trebamo uključiti slijedeće zaglavlje:
    1. pthread.h
12. Koliko imamo različitih tipova događaja (event). Tip događaja može se zadati prilikom kreiranja s CreateEvent funkcijom.
    1. 2
13. Posebna funkcija koja postoji kod pniti pod nazivom sched\_yield radi sljedeće:
    1. nit ne treba više procesor i prelazi u stanje ‘spremna’
14. Je li navedeno ime imenovane pipe korektno? (Ovo ime klijent predaje funkciji za kreiranje HANDLE-a na imenovanu pipu kako bi mogao komunicirati sa serverom)
    1. \\.\pipe\mojapipa → korektno
    2. \\\*\pipe\mojapipa → nije korektno
    3. \\server1\mojapipa → nije korektno
    4. mojapipa → nije korektno
    5. \\officepc\pipe\mojapipa → korektno
15. Možemo čekati na 1 ili više HANDLE-ova mutex-a s funkcijama:
    1. WaifForMultipleObjects
    2. WaitForSingleObject
16. Prilikom kreiranja servera koji koristi imenovanje pipe ‘kratkog života’ redoslijed pozivanja funkcija mora biti: (povežite vremenski redoslijed pozivanja funkcija)
    1. CreateNamedPipe → 1
    2. ConnectNamedPipe → 2
    3. ReadFile/WriteFile → 3
    4. DisconnectNamedPipe → 4
    5. CloseHandle → 5
17. Kako se naziva mehanizam za sinkronizaciju koji je dobio ime po skraćenici svoje namjene: međusobno isključivanje.
    1. Mutex
18. Napišite kraticu koja označava rad s nitima koje smo označili kao p-niti (pthreads). Ta kratica počinje sa slovom p.
    1. POSIX

Velika skripta: [SP - Skripte.pdf - Google Drive](https://drive.google.com/file/d/13zyhCKWrgaV6m98k4Vl-hrYPM_oON8kt/view)

Sve prezentacije za test3 (SP\_test3.pdf): [Sistemsko Programiranje - Google Drive](https://drive.google.com/drive/folders/1sVDex7u26X2Etztz6KaKiGxu9mWDjvFt?fbclid=IwAR2U_P2d_Vn6RRhefQgIqFeSVdKf-BCsXmDlWyDICQaUZ-gDg659bhvnklg)