Լաբորատոր աշխատանք 7

Սոկետներ (Sockets)
UNIX դոմեյն

Սոկետը պրոցեսների հաղորդակցության (IPC) մեթոդ է, որը հնարավորություն է տալիս ծրագրերի միջև կատարել տվյալների փոխանակում: Տվյալների փոխանակում հնարավոր է կատարել և՛ նույն հոսթի (համակարգչի) վրա աշխատող ծրագրերի համար, և՛ տարբեր հոսթերի միջև:

Սոկետներն ունեն հաղորդակցության դոմեյն, որը սահմանում է հաղորդակցության տիրույթը (նույն հոսթի վրա գտնվող ծրագրեր կամ ցանցով միմյանց կապակցված տարբեր հոսթերի ծրագրեր): Հաճախ կիրառվող դոմեյններ են.

- **UNIX (AF_UNIX)** դոմեյնը թույլ է տալիս նույն հոսթի վրա գտնվող ծրագրերի հաղորդակցություն:
- **IPv4 (AF_INET)** դոմեյնը թույլ է տալիս IPv4 (Internet Protocol version 4) արձանագրությամբ միմյանց միացված հոսթերի ծրագրերի հաղորդակցություն:
- **IPv6 (AF_INET6)** դոմեյնը թույլ է տալիս IPv6 (Internet Protocol version 6) արձանագրությամբ միմյանց միացված հոսթերի ծրագրերի հաղորդակցություն:

Սոկետների յուրաքանչյուր իրականացում ունի առնվազն 2 տիպի սոկետ.

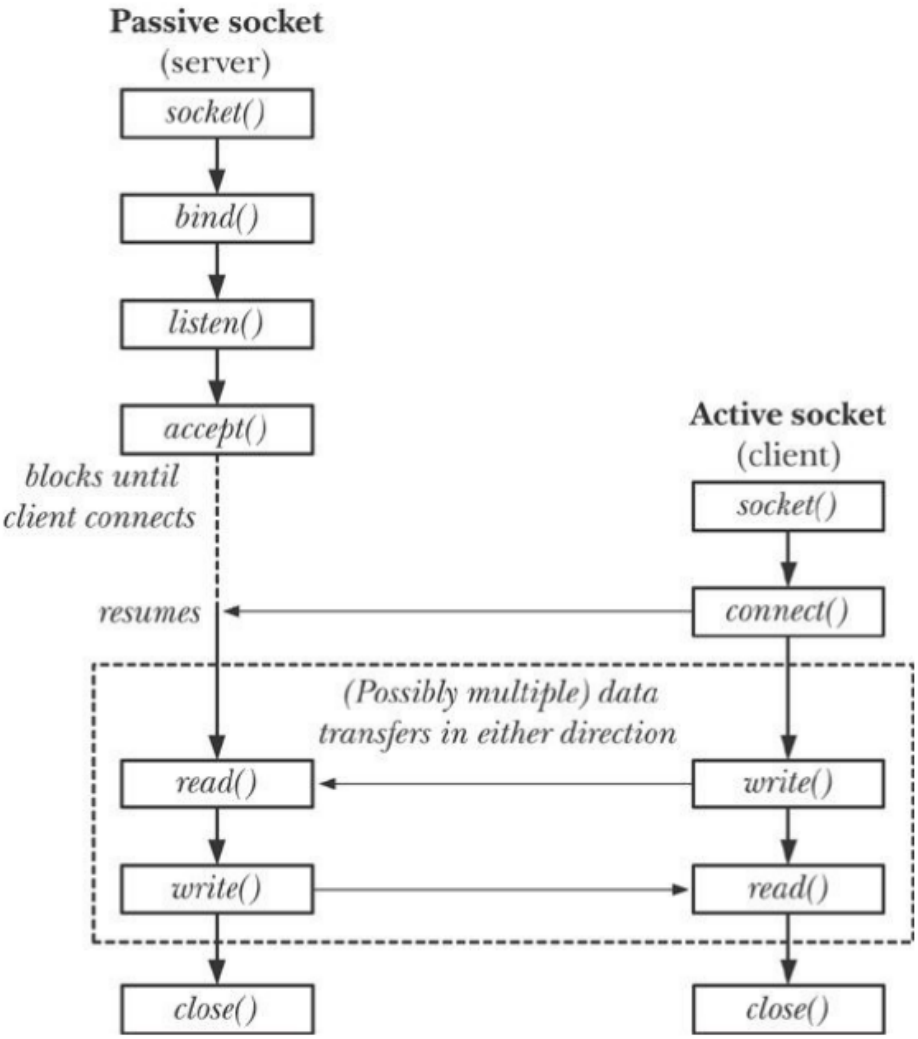
- **Հոսքային (SOCK_STREAM)** – ապահովում է հուսալի, երկկողմանի, բայթային հոսքով տվյալների փոխանակում: Աշխատանքի համար պահանջում է մեկ այլ սոկետի հետ միացում:
- **Դեյտագրամային (SOCK_DGRAM)** – ապահովում է տվյալների փոխանակումը դեյտագրամների (հաղորդագրություն, փաթեթ) տեսքով: Այս եղանակով տվյալների փոխանակումը հուսալի չէ. հաղորդագրությունները կարող են չուղարկվել, կրկնօրինակվել կամ կորչել: Այլ սոկետի հետ միացումը պարտադիր չէ:

Հոսքային սոկետներ (Stream Sockets)

Հոսքային սոկետների աշխատանքի սկզբունքը ցուցադրված է Նկ. 1-ում:

- Կողմերից յուրաքանչյուրը ստեղծում է նոր սոկետ՝ կանչելով **socket()** ֆունկցիան:
- Կողմերից մեկը կանչում է **bind()** ֆունկցիան, որպեսզի միացնի իր սոկետը հայտնի հասցեին (well-known address): Դրանից հետո այն կանչում է **listen()** ֆունկցիան՝ տեղեկացնելով ՕՀ միջուկին, որ պատրաստ է ընդունել մուտքային կապեր:
- Մյուս կողմը կապ է հաստատում ՝ կանչելով **connect()** ֆունկցիան և սահմանելով հասցեն, որով պետք է միացում իրականացվի:

- Առաջին կողմը, որը կանչել էր **listen()**, այնուհետև ընդունում է կապը՝ կանչելով **accept()** ֆունկցիան:
- Կապը հաստատվելուց հետո տվյալները կարող են փոխանցվել 2 ուղղություններով, մինչև կողմերից մեկը փակի կապը՝ **close()** կանչի միջոցով:



Նկ. 1 Հոսքային սոկետների աշխատանքը

unix_stream_server, unix_stream_client ծրագրերը

Այս ծրագրերը ներկայացնում են UNIX դոմեյնում հոսքային սոկետների աշխատանքը: Client ծրագիրը միանում է server ծրագրին և ուղարկում է STDIN հոսքից ստացված տվյալները: Server ծրագիրը ստանում է այս տվյալները և արտածում:

Server ծրագիրն իրականացնում է հետևյալ քայլերը.

- Ստեղծում է սոկետ:

- Հեռացնում է /tmp/unix_stream ֆայլը, եթե այն գոյություն ունի: Այս ֆայլը ծրագրում օգտագործվում է որպես հայտնի հասցե (well-known address), որին պետք է կապվի սոկետը՝ bind() կանչի միջոցով: Այս քայլն անհրաժեշտ է իրականացնել, քանի որ հայտնի հասցեն կարող է կապվել միայն մեկ սոկետի հետ:
- Կառուցում է սոկետի հասցեի ստրուկտուրան, կապում է սոկետը այդ հասցեին՝ կանչելով bind() ֆունկցիան և սոկետը դարձնում է լսող (պասիվ)՝ կանչելով listen() ֆունկցիան:
- Կատարում է անվերջ ցիկլ, որում մշակում է client ծրագրից ստացված միացման հարցումները: Ցիկլի յուրաքանչյուր իտերացիայում կատարում է հետևյալ քայլերը.
    - Ընդունում է կապը, որի արդյունքում ստեղծվում է նոր սոկետ՝ cfd:
    - cfd սոկետից կարդում է բոլոր տվյալները և արտածում դրանք:
    - Փակում է cfd սոկետը:

  Client ծրագիրն իրականացնում է հետևյալ քայլերը.

- Ստեղծում է սոկետ:
- Ստեղծում է server-ի սոկետի հասցեի ստրուկտուրան և միանում այդ հասցեի սոկետին:
- Կատարում է ցիկլ, որում պատճենում է STDIN հոսքից ստացված տվյալները սոկետի միացման մեջ: Ծրագիրն ավարտվում է EOF սիմվոլ ստանալիս (Ctrl + D):

Ծրագիրը կատարելու համար անհրաժեշտ քայլերն են.

- Ստեղծել ծրագրերի կատարվող ֆայլերը.
  gcc unix_stream_server.c -o stream_server
  gcc unix_stream_client.c -o stream_client
- Կատարել ծրագրերն առանձին հրամանային տողերում.
  ./stream_server
  ./stream_client

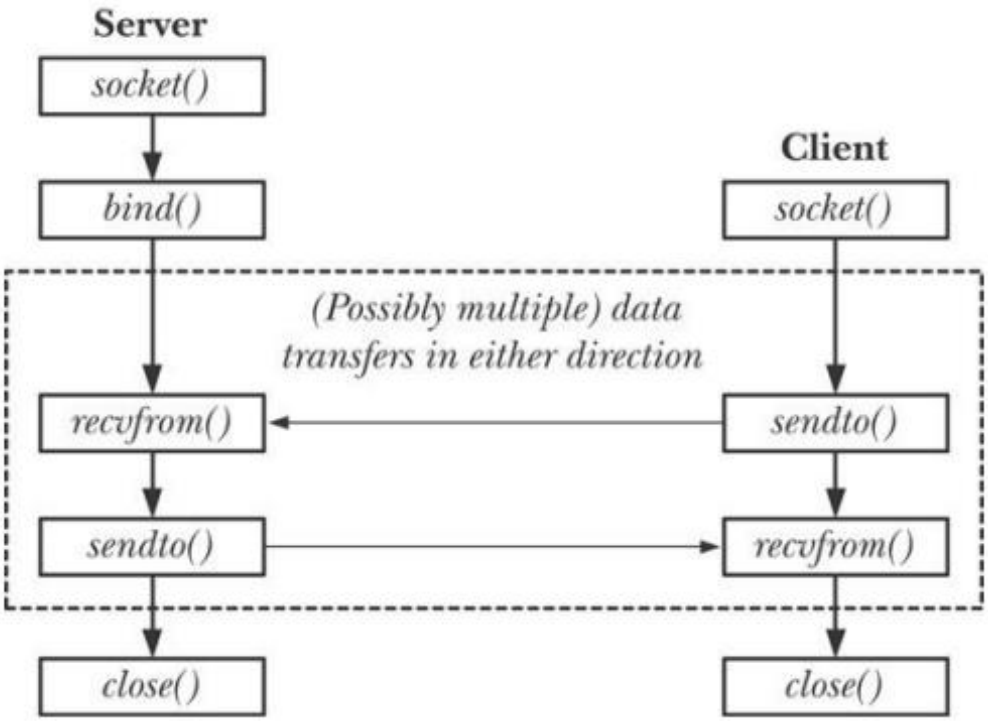
Դեյտագրամային սոկետներ (Datagram Sockets)

Դեյտագրամային սոկետների աշխատանքի սկզբունքը ցուցադրված է Նկ. 2-ում:

- Կողմերից յուրաքանչյուրը ստեղծում է նոր սոկետ՝ կանչելով **socket()** ֆունկցիան:
- Կողմերից մեկը կանչում է **bind()**՝ կապելով իր սոկետը հայտնի հասցեի, որպեսզի մեկ այլ ծրագիր կարողանա հաղորդագրություններ ուղարկել:
- Հաղորդագրություն ուղարկելու համար ծրագիրը կանչում է **sendto()** ֆունկցիան, որի արգումենտներից մեկն այն սոկետի հասցեն է, որին պետք է ուղարկվի հաղորդագրությունը:
- Հաղորդագրությունը ստանալու նպատակով առաջին ծրագիրը կանչում է **recvfrom()** ֆունկցիան, որը կարող է արգելափակվել, եթե հաղորդագրությունը դեռ

չի ստացվել: Քանի որ recvfrom() ֆունկցիան թույլ է տալիս ստանալ ուղարկողի հասցեն, ապա կարելի է ցանկության դեպքում պատասխանել:

- Երբ սոկետն այլևս հարկավոր չէ, ծրագիրը փակում է այն՝ կաչելով *close()* ֆունկցիան:



Նկ. 2 Դեյտագրամային սոկետների աշխատանքը

unix_datagram_server, unix_datagram_client ծրագրերը

Այս ծրագրերը ներկայացնում են Unix դոմեյնում դեյտագրամային սոկետների աշխատանքը: Client ծրագիրը server-ին ուղարկում է STDIN հոսքից ստացված տվյալները, server-ը դրանք դարձնում է մեծատառ և վերադարձնում client-ին: Ստացված արդյունքն արտածվում է:

Server ծրագիրն իրականացնում է հետևյալ քայլերը.

- Ստեղծում է սոկետ:
- Հեռացնում է /tmp/unix_datagram ֆայլը, եթե այն գոյություն ունի: Այս ֆայլը server ծրագրում օգտագործվում է որպես հայտնի հասցե:
- Կառուցում է սոկետի հասցեի ստրուկտուրան, կապում է սոկետը այդ հասցեին՝ կանչելով bind() ֆունկցիան

- Կատարում է անվերջ ցիկլ, որում մշակում է client ծրագրից ստացված միացման հարցումները: Ցիկլի յուրաքանչյուր իտերացիայում կատարում է հետևյալ քայլերը.
    - recvfrom() ֆունկցիայի միջոցով ստանում է client-ի կողմից ուղարկված հաղորդագրությունը և client-ի հասցեն:
    - Ստացված հաղորդագրությունը դարձնում է մեծատառ և ուղարկում client-ին:

Client ծրագիրն իրականացնում է հետևյալ քայլերը.

- Ստեղծում է սոկետ:
- Կապում է սոկետը որևէ հասցեի, որպեսզի server-ը կարողանա պատասխան հաղորդագրություն ուղարկել: Հասցեն եզակի դարձնելու համար /tmp/unix_datagram ֆայլի անվան վերջում կցվում է պրոցեսի id-ն:
- Ուղարկում է հրամանային տողից ստացված արգումենտներից յուրաքանչյուրը սերվերին:
- Կարդում է սերվերի կողմից ստացված պատասխանը և արտածում:

Ծրագիրը կատարելու համար անհրաժեշտ քայլերն են.

- Ստեղծել ծրագրերի կատարվող ֆայլերը.
  gcc unix_datagram_server.c -o dg_server
  gcc unix_datagram_client.c -o dg_client
- Կատարել ծրագրերն առանձին հրամանային տողերում.
  ./dg_server
  ./dg_client hello

Առաջադրանքներ

1. Կատարել unix_stream_server և unix_stream_client ծրագրերը:
2. unix_stream_server ծրագրից հեռացնել /tmp/unix_stream ֆայլը ջնջող հատվածը: Կրկին կատարել ծրագիրը և ցուցադրել ստացված սխալը:
3. Կատարել unix_datagram_server և unix_datagram_client ծրագրերը: unix_datagram_client ծրագրին փոխանցել 'long message' տողը, ցուցադրել ստացված արդյունքը:
4. Հոսքային սոկետների միջոցով իրականացնել Echo service: Server ծրագիրը պետք է միացումներ ընդունի տարբեր client-ներից և հետ ուղարկի ստացված տվյալները:
5. Հոսքային սոկետների միջոցով իրականացնել chatroom: Server ծրագիրը պետք է միացումներ ընդունի տարբեր client-ներից և դրանցից յուրաքանչյուրի կողմից ստացված հաղորդագրությունը ուղարկի մյուս client-ներին: