Proiect SCA

Comunicatia prin interfata CAN

Realizatori:

Borlea Anamaria-Ioana

Dumitru Laurentiu

**1.Implementarea cerintelor proiectului 1**

Create a gateway which offers the following options:

* Select the number of interfaces

    printf("Introdu numarul de interfete dorit.\n");

    scanf("%d", &number\_of\_interfaces);

    int\*\* typeList = (int\*\*)malloc(number\_of\_interfaces \* sizeof(int\*));

    for(int i = 0; i < number\_of\_interfaces; i++ ) {

    typeList[i] = (int\*)malloc(number\_of\_interfaces \* sizeof(int));

    }

* Configure the type of each interface(CAN,CAN FD,VCAN) and bitrate.

for ( int i = 0; i < number\_of\_interfaces; i++ )

        {

            do{

            printf("Alege tipul de interfata pentru interfata %d : \n 1.CAN \n 2.CAN FD \n 3.VCAN \n", i+1);

            scanf("%d", &tipul\_interfetei);

            typeList[i][i] = tipul\_interfetei;

            }while(tipul\_interfetei > 3 && tipul\_interfetei < 0);

        }

        printf("Introdu bitrate-ul.\n");

        scanf("%d", &bitrate);

* Select from menu what interface should be made active/speed/ type - CAN, CANFD

    int\* activeList = malloc(number\_of\_interfaces \* sizeof(int));

    for (int i = 0; i < number\_of\_interfaces; i++)

    {

        printf("Seteaza starea de lucru a interfetei %d. (0. Oprit, 1. Pornit) \n", i+1);

        scanf("%d", &workingState);

        activeList[i] = workingState;

    }

Deasemea am implementat optiunea de a active/dezactiva o interfata la orice moment in timpul rularii:

void control\_interface(int chosen\_interface, int bitrate) // Functie folosita pentru oprirea sau pornirea interfetei alese de utilizator.

{

    char buffer[25], buffer2[50];

    char command[150] = "sudo ip link set vcan";

    int choice;

    printf("Do you want to shutdown or to activate the interface?\n");

    printf("1.Shutdown. 2.Activate.\n");

    scanf("%d", &choice);

    if(choice == 1){

          activeList[chosen\_interface-1] = 0;

          sprintf(buffer, "%d", choice);

          strcat(command, buffer);

          strcat(command, " down");

          system(command);

    }else{

          activeList[chosen\_interface-1] = 1;

          sprintf(buffer, "%d", choice);

          strcat(command, buffer);

          strcat(command, " up type can bitrate ");

          sprintf(buffer2, "%d", bitrate);

          strcat(command, buffer2);

          system(command);

          printf("%s", command);

    }

}

* Configure the routing rules

for ( int i = 0; i < number\_of\_interfaces; i++ )

        {

            do{

            for(int j =0;j<number\_of\_interfaces;j++)

            {

                if(i!=j)

                {

                    printf("Alege regilile de rutare pentru interfata %d raportat la interfata %d: \n 0.Nu trimite \n 1.Trimite \n", i+1,j+1);

                    scanf("%d", &tipul\_interfetei);

                    typeList[i][j] = tipul\_interfetei;

                }

            }

            }while(tipul\_interfetei > 3 && tipul\_interfetei < 0);

        }

* Add remove filters

void change\_filters(int number\_of\_interfaces, int bitrate)  // schimbarea rutelor de comunicare intre interfete

{

    int chosen\_interface\_send, chosen\_interface\_receive;

    char buffer[10], buffer2[30], buffer3[10];

    char command[150] = "sudo ip link set vcan";

    char routingCall[100] = "sudo cangw -A -s vcan";

    printf("\nChoose the interface that you want to send data\n");

    scanf("%d", &chosen\_interface\_send);

    printf("Choose the interface that you want to receive data\n");

    scanf("%d", &chosen\_interface\_receive);

    if ((chosen\_interface\_send != chosen\_interface\_receive) &&

        (chosen\_interface\_send <= number\_of\_interfaces && chosen\_interface\_send > 0) &&

        (chosen\_interface\_receive <= number\_of\_interfaces && chosen\_interface\_receive > 0))

    {

          printf("Enabling chosen interfaces....\n");

          activeList[chosen\_interface\_send-1] = 1;

          activeList[chosen\_interface\_receive-1] = 1;

          sprintf(buffer, "%d", chosen\_interface\_send-1);

          strcat(command, buffer);

          strcat(command, " up type can bitrate ");

          sprintf(buffer2, "%d", bitrate);

          strcat(command, buffer2);

          system(command); // Activarea interfetei care genereaza date.

          printf("\n%s\n", command);

          strcpy(command, "sudo ip link set vcan");

          sprintf(buffer, "%d", chosen\_interface\_receive-1);

          strcat(command, buffer);

          strcat(command, " up type can bitrate ");

          sprintf(buffer2, "%d", bitrate);

          strcat(command, buffer2);

          system(command); // Activarea interfetei care primeste date.

          printf("\n%s\n", command);

          printf("Establishing routing rules...\n");

          sprintf(buffer3, "%d", chosen\_interface\_send-1);

          strcat(routingCall, buffer3);

          strcat(routingCall, " -d vcan");

          sprintf(buffer3, "%d", chosen\_interface\_receive-1);

          strcat(routingCall, buffer3);

          strcat(routingCall, " -e");

          system(routingCall); // setarea rutei de comunicare

          printf("\n%s\n", routingCall);

          typeList[chosen\_interface\_send-1][chosen\_interface\_receive-1] = 1; // setarea directiei de comunicare a interfetei ( rand - interfata, coloana - interfata cu care comunica )

    }else

    printf("Choose a valid interface!\n");

}

* Print each CAN frame (ID, DLC,data, in interface, out interface/interfaces)

void print\_frames(int number\_of\_interfaces,int bitrate)

{

    system("canplayer -I \"$(ls -t | head -n1)\" -v"); // print al ultimelor frame-uri generate

}

Pentru o buna functionalitate trebuie ca inainte de trimiterea unui frame sa existe o interfata de can care asculta bus-ul, in cazul aplicatiei noastre, este necesara rularea in prealabil a programului secondProgram.c .

int main()

{

    int choice;

    char candumpCommand[30] = "candump -l vcan";

    char buffer[10];

    do{

    printf("Select the interface on which the data dump will be made!\n");

    printf("Type in 9 to stop! Close the dump when the generation of data finished! ( CTRL + C )\n");

    scanf("%d", &choice);

    if ( choice != 9 )

    {

        sprintf(buffer, "%d", choice-1);

        strcat(candumpCommand, buffer);

        system(candumpCommand);

    }else

        printf("Exiting...\n");

    }while(choice != 9);

}

**2.Testarea aplicatiei**

Pentru a putea fi testate ma usor aplicatia, am implementat doua variante de a o executa: cu scenario de test predefinite sau cu citirea datelor introduce de utilizator in consola, lucru exemplificat mai sus.

    ///////// HERE ARE CONFIGURATIONS ALREADY MADE FOR TESTING PURPOSES /////////

    do{

    printf("Choose one configuration: \n 1.VCAN config (only working for some reason)\n 2.CAN config \n 3.CAN FD config \n 4.CAN FD+CAN config\n");

    scanf("%d", &configuration\_number);

    if(configuration\_number == 1)

    {

        number\_of\_interfaces=3;

        typeList[0][0]=3;

        typeList[1][1]=3;

        typeList[2][2]=3;

        typeList[0][1]=1;

        typeList[0][2]=0;

        typeList[1][0]=0;

        typeList[1][2]=1;

        typeList[2][0]=1;

        typeList[2][1]=1;

        bitrate=25000;

        activeList[0]=1;

        activeList[1]=1;

        activeList[2]=1;

    //pe diagonala principala e trecut tipul de can 1.CAN; 2.CAN FD; 3.VCAN;

    //0 inseamna ca nu trimite si 1 inseamna ca trimite la interfata cu indicele corespunzator

    }else if(configuration\_number == 2)

    {

        number\_of\_interfaces=3;

        //typeList={1, 1, 0, 0, 1, 1, 1, 1, 1};

        bitrate=50000;

        //activeList[3]={1, 1, 1};

    }else if(configuration\_number == 3)

    {

        number\_of\_interfaces=3;

        //typeList={2, 1, 0, 0, 2, 1, 1, 1, 2};

        bitrate=4000000;

        //activeList[3]={1, 1, 1};

    }else if(configuration\_number == 4){

        number\_of\_interfaces=3;

        //typeList={1, 1, 0, 0, 2, 1, 1, 1, 1};

        bitrate=25000;

        //activeList[3]={1, 1, 1};

    }

    }while(configuration\_number >  3 && configuration\_number < 1);