***EX:3 PRIORITY & RR***

***CPU SCHEDULING***

***-S.Vishakan CSE-C 18 5001 196***

***Source Code – (Uppercase – Parent & Child):***

**#include <sys/ipc.h>**

**#include <sys/shm.h>**

**#include <sys/types.h>**

**#include <sys/wait.h>**

**#include <stdio\_ext.h>**

**#include <unistd.h>**

**#include <stdio.h>**

**#include <stdlib.h>**

**#include <string.h>**

**#include <ctype.h>**

**char \*upperCase(char \*word); //function for converting to uppercase**

**int main(void){**

**int pid, id;**

**char \*l\_word, \*u\_word;**

**id = shmget(111, 1024, IPC\_CREAT|00666); //identifier for shared memory**

**pid = fork(); //child process and parent process share the same memory**

**if(pid > 0){**

**l\_word = (char\*) shmat(id,(void\*)0,0); //attaching to shared memory**

**printf("\nEnter a Word: ");**

**fgets(l\_word, 1000, stdin);**

**wait(0);**

**shmdt(l\_word); //detaching from shared memory after placing the word**

**}**

**else{**

**sleep(7); //waiting for 7 seconds when the writer process is putting words into shared memory**

**u\_word = (char\*) shmat(id,(void\*)0,0); //attaching once writer is complete**

**printf("Received word : %s",upperCase(u\_word));**

**shmdt(u\_word); //detach**

**exit(0);**

**}**

**}**

**char \*upperCase(char \*word){**

**int len = strlen(word);**

**int i = 0;**

**char \*uword;**

**uword = (char \*)malloc(sizeof(char)\*len);**

**for(;i<len;i++)**

**uword[i] = toupper(word[i]);**

**return uword;**

**}**

***/\****

***OUTPUT:***

***(base) vishakan@Legion:~/Desktop/Operating-Systems/Ex4 InterProcess Communication$ gcc 1-Uppercase.c -o u***

***(base) vishakan@Legion:~/Desktop/Operating-Systems/Ex4 InterProcess Communication$ ./u***

***Enter a Word: interprocess communication***

***Received word : INTERPROCESS COMMUNICATION***

***\*/***

***Source Code*** ***– (Sender Program):***

**#include <sys/ipc.h>**

**#include <sys/shm.h>**

**#include <sys/types.h>**

**#include <sys/wait.h>**

**#include <stdio\_ext.h>**

**#include <unistd.h>**

**#include <stdio.h>**

**#include <stdlib.h>**

**#include <string.h>**

**#include <ctype.h>**

**int main(void){**

**int id;**

**char \*msg;**

**id = shmget(111, 1024, IPC\_CREAT|00666);**

**msg = shmat(id, NULL, 0);**

**printf("Enter the Message to be Transferred: ");**

**scanf("%[^\n]",msg);**

**shmdt(msg); //sending message through shm and detaching**

**sleep(0);**

**exit(0);**

**}**

***/\****

***OUTPUT:***

***(base) vishakan@Legion:~/Desktop/Operating-Systems/Ex4 InterProcess Communication$ gcc 2-Sender.c -o s***

***(base) vishakan@Legion:~/Desktop/Operating-Systems/Ex4 InterProcess Communication$ ./s***

***Enter the Message to be Transferred: Shared Memory Check, One...Two...Three!***

***(base) vishakan@Legion:~/Desktop/Operating-Systems/Ex4 InterProcess Communication$ gcc 2-Receiver.c -o r***

***(base) vishakan@Legion:~/Desktop/Operating-Systems/Ex4 InterProcess Communication$ ./r***

***Received the following message:***

***Shared Memory Check, One...Two...Three!***

***\*/***

***Source Code – (Receiver Program):***

**#include <sys/ipc.h>**

**#include <sys/shm.h>**

**#include <sys/types.h>**

**#include <sys/wait.h>**

**#include <stdio\_ext.h>**

**#include <unistd.h>**

**#include <stdio.h>**

**#include <stdlib.h>**

**#include <string.h>**

**#include <ctype.h>**

**int main(void){**

**int id;**

**char \*msg;**

**id = shmget(111, 1024, IPC\_CREAT|00666);**

**msg = shmat(id, NULL, 0);**

**printf("\nReceived the following message:\n%s\n",msg);**

**shmdt(msg);**

**shmctl(id, IPC\_RMID, NULL); //destroying the shared memory contents**

**sleep(0);**

**exit(0);**

**}**

***/\****

***OUTPUT:***

***(base) vishakan@Legion:~/Desktop/Operating-Systems/Ex4 InterProcess Communication$ gcc 2-Sender.c -o s***

***(base) vishakan@Legion:~/Desktop/Operating-Systems/Ex4 InterProcess Communication$ ./s***

***Enter the Message to be Transferred: Shared Memory Check, One...Two...Three!***

***(base) vishakan@Legion:~/Desktop/Operating-Systems/Ex4 InterProcess Communication$ gcc 2-Receiver.c -o r***

***(base) vishakan@Legion:~/Desktop/Operating-Systems/Ex4 InterProcess Communication$ ./r***

***Received the following message:***

***Shared Memory Check, One...Two...Three!***

***\*/***

***Source Code – (Peer 1 Program):***

**#include <sys/ipc.h>**

**#include <sys/shm.h>**

**#include <sys/types.h>**

**#include <sys/wait.h>**

**#include <stdio\_ext.h>**

**#include <unistd.h>**

**#include <stdio.h>**

**#include <stdlib.h>**

**#include <string.h>**

**#include <ctype.h>**

**int main(void){**

**int id;**

**char msg[100], \*buffer;**

**char temp1[100], temp2[100];**

**printf("\n\t\t\t\tP2P Chat\n\n");**

**id = shmget(111, 1024, IPC\_CREAT|00666); //Opening a shared memory access**

**buffer = shmat(id, NULL, 0); //Attaching a buffer to it**

**printf("\n\n\t\tThe Chat Connection has been Opened!.\n\t\t\tEnter \"Bye\" to Quit.\n");**

**printf("\nYou:\n\t");**

**fgets(temp1, 100, stdin);**

**strcpy(msg, "~"); //Clearing the strings with a preset ~ value**

**strcat(msg, temp1);**

**strcpy(buffer, msg);**

**strcpy(msg, "~");**

**while(1){**

**strcpy(msg, "~"); //Clearing the msg buffer with ~ again for next time**

**while(buffer[0] == '~'); //Waiting, if the buffer is empty.**

**strcpy(temp2, buffer);**

**char \*sep = strtok(temp2, "`"); //Splitting the string at the preset value ` for the other client**

**printf("\nPeer:\n\t%s\n", sep);**

**if(strcmp(sep, "Bye\n") == 0){ //Ending the chat if "Bye" is entered by the other user.**

**break;**

**}**

**else{**

**printf("You:\n\t");**

**fgets(temp1, 100, stdin);**

**strcat(msg, temp1); //Putting the scanned value into buffer**

**strcpy(buffer, msg); //Now buffer is like ~<msg>**

**strcat(msg, "~"); //Now msg is like ~<msg>~**

**if(strcmp(temp1, "Bye\n") == 0){ //Exiting the chat this user enters "Bye"**

**break;**

**}**

**}**

**}**

**printf("\n\n\t\tThe Chat Connection has been Closed.\n");**

**shmdt(buffer);**

**shmctl(id, IPC\_RMID, NULL); //Deleting the shared memory addressing**

**sleep(0);**

**exit(0);**

**}**

***/\****

***OUTPUT:***

***(base) vishakan@Legion:~/Desktop/Operating-Systems/Ex4 InterProcess Communication$ gcc 3-ChatPeer1.c -o c1***

***(base) vishakan@Legion:~/Desktop/Operating-Systems/Ex4 InterProcess Communication$ ./c1***

***P2P Chat***

***The Chat Connection has been Opened!.***

***Enter "Bye" to Quit.***

***You:***

***Hey***

***Peer:***

***Hello***

***You:***

***How's it going?***

***Peer:***

***It's going good!***

***You:***

***Good to know :)***

***Peer:***

***Bye***

***You:***

***Bye***

***The Chat Connection has been Closed.***

***\*/***

***Source Code – (Peer 2 Program):***

**#include <sys/ipc.h>**

**#include <sys/shm.h>**

**#include <sys/types.h>**

**#include <sys/wait.h>**

**#include <stdio\_ext.h>**

**#include <unistd.h>**

**#include <stdio.h>**

**#include <stdlib.h>**

**#include <string.h>**

**#include <ctype.h>**

**int main(void){**

**int id;**

**char msg[100], \*buffer;**

**char temp1[100], temp2[100];**

**printf("\n\t\t\t\tP2P Chat\n\n");**

**id = shmget(111, 1024, IPC\_CREAT|00666); //Opening a shared memory access**

**buffer = shmat(id, NULL, 0); //Attaching a buffer to it**

**printf("\n\n\t\tThe Chat Connection has been Opened!.\n\t\t\tEnter \"Bye\" to Quit.\n");**

**strcpy(msg, "`"); //Clearing the strings with a preset ` value**

**strcpy(buffer, "`");**

**while(1){**

**strcpy(msg, "`"); //Clearing the msg buffer with ` again for next time**

**while(buffer[0] == '`'); //Waiting, if the buffer is empty.**

**strcpy(temp2, buffer);**

**char \*sep = strtok(temp2, "~"); //Splitting the string at the preset value ~ for the other client**

**printf("\nPeer:\n\t%s\n", sep);**

**if(strcmp(sep, "Bye\n") == 0){ //Ending the chat if "Bye" is entered by the other user.**

**break;**

**}**

**else{**

**printf("You:\n\t");**

**fgets(temp1, 100, stdin);**

**strcat(msg, temp1); //Putting the scanned value into buffer**

**strcpy(buffer, msg); //Now buffer is like `<msg>**

**strcat(msg, "`"); //Now msg is like `<msg>`**

**if(strcmp(temp1, "Bye\n") == 0){ //Exiting the chat this user enters "Bye"**

**break;**

**}**

**}**

**}**

**printf("\n\n\t\tThe Chat Connection has been Closed.\n");**

**shmdt(buffer);**

**shmctl(id, IPC\_RMID, NULL); //Deleting the shared memory addressing**

**sleep(0);**

**exit(0);**

**}**

***/\****

***OUTPUT:***

***(base) vishakan@Legion:~/Desktop/Operating-Systems/Ex4 InterProcess Communication$ gcc 3-ChatPeer2.c -o c2***

***(base) vishakan@Legion:~/Desktop/Operating-Systems/Ex4 InterProcess Communication$ ./c2***

***P2P Chat***

***The Chat Connection has been Opened!.***

***Enter "Bye" to Quit.***

***Peer:***

***Hey***

***You:***

***Hello***

***Peer:***

***How's it going?***

***You:***

***It's going good!***

***Peer:***

***Good to know :)***

***You:***

***Bye***

***The Chat Connection has been Closed.***

***\*/***