**Shared memory:**

**Shmget()-** is used to create the shared memory segment

#include<sys/ipc.h>

#include<sys/shm.h>

int shmget(key\_t key, size\_t size, int shmflg);

*key- unique number used to identify the shared segment (as multiple shared segments can be created)*

*size- size of shared memory in bytes*

*shmflg- gives permission to the shared memory*

**Returns—**(1) If successful, shmget returns a nonnegative integer corresponding to the shared memory segment identifier.

(2) If unsuccessful, shmget returns -1 and sets errno

Eg. shmget(key,20,0664|IPC\_CREAT);

**Shmat()-** is used to attach the shared segment with the address space of the process.

#include<sys/ipc.h>

#include<sys/shm.h>

int shmget(key\_t key, size\_t size, int shmflg);

eg. shmat(shmid,NULL,0);

#include<stdio.h>

#include<stdlib.h>

#include<unistd.h>

#include<sys/shm.h>

#include<string.h>

#include<sys/wait.h>

int main(){

pid\_t pid;

int shmid,\*shvar;

key\_t key=ftok(".",45);

shmid=shmget(key,20,0664|IPC\_CREAT);

printf("Key=%x ........Shmid=%d\n",key,shmid);

shvar=shmat(shmid,NULL,0);

printf("Default initial value of shvar=%d\n",\*shvar);

\*shvar=10;

pid=fork();

if(pid==0){

\*shvar=\*shvar+90;

printf("child update=%d\n",\*shvar);

exit(0);

}

else{

wait(NULL);

\*shvar=\*shvar+110;

printf("parent updates=%d\n",\*shvar);

}

//shmdt(shvar);

//shmctl(shmid,IPC\_RMID,NULL);

return 0;

}

Output:--

Key=2d8c5c32 ........Shmid=0

Default initial value of shvar=0

child update=100

parent updates=210

#include<stdio.h>

#include<stdlib.h>

#include<unistd.h>

#include<sys/shm.h>

#include<string.h>

#include<sys/wait.h>

int main(){

pid\_t pid;int shmid,\*shvar;key\_t key=ftok(".",45);

shmid=shmget(key,20,0664|IPC\_CREAT);

printf("Key=%x ........Shmid=%d\n",key,shmid);

shvar=shmat(shmid,NULL,0);

printf("Default initial value of shvar=%d\n",\*shvar);

\*shvar=10;pid=fork();

if(pid==0){

\*shvar=\*shvar+90;

printf("child update=%d\n",\*shvar);

exit(0);

}

else{

\*shvar=\*shvar+110;

printf("parent updates=%d\n",\*shvar);

wait(NULL);

}

//shmdt(shvar);

//shmctl(shmid,IPC\_RMID,NULL);

return 0;

}

Output:

Key=2d8c5c32 ........Shmid=0

Default initial value of shvar=210

parent updates=120

child update=210

#include<stdio.h>

#include <sys/types.h>

#include <sys/ipc.h>

#include<sys/shm.h>

int main()

{

int id,\*var;

key\_t key;

key=ftok("key.txt",65);

id=shmget(key,50,0664|IPC\_CREAT);

printf("Shared memory Identifier=%d\n",id);

var=(int \*)shmat(id, NULL,0);

\*var=50;

shmdt(var);

return 0;

}

Sm1r.c

#include <sys/ipc.h>

#include <sys/shm.h>

#include <stdio.h>

int main()

{

// ftok to generate unique key

key\_t key = ftok("shmfile",65);

// shmget returns an identifier in shmid

int shmid = shmget(key,1024,0666|IPC\_CREAT);

// shmat to attach to shared memory

char \*str = (char\*) shmat(shmid,(void\*)0,0);

printf("Data read from memory: %s\n", (char\*)str);

//detach from shared memory

shmdt(str);

// destroy the shared memory

shmctl(shmid,IPC\_RMID,NULL);

return 0;

}

#include <sys/ipc.h>

#include <sys/shm.h>

#include <stdio.h>

int main()

{

// ftok to generate unique key

key\_t key = ftok("shmfile",65);

// shmget returns an identifier in shmid

int shmid = shmget(key,1024,0666|IPC\_CREAT);

// shmat to attach to shared memory

char \*str = (char\*) shmat(shmid,(void\*)0,0);

printf("Write Data : ");

gets(str);

printf("Data written in memory: %s\n",str);

//detach from shared memory

shmdt(str);

return 0;

}

#include<stdio.h>

#include<stdlib.h>

#include<unistd.h>

#include<sys/shm.h>

#include<string.h>

int main()

{

int i;

void \*shared\_memory;

char buff[100];

int shmid;

shmid=shmget((key\_t)2345, 1024, 0666);

printf("Key of shared memory is %d\n",shmid);

shared\_memory=shmat(shmid,NULL,0); //process attached to shared memory segment

printf("Process attached at %p\n",shared\_memory);

printf("Data read from shared memory is : %s\n",(char \*)shared\_memory);

}

#include<stdio.h>

#include<stdlib.h>

#include<unistd.h>

#include<sys/shm.h>

#include<string.h>

int main()

{

int i;

void \*shared\_memory;

char buff[100];

int shmid;

shmid=shmget((key\_t)2345, 1024, 0666|IPC\_CREAT); //return value is the identifier of shared memory

//creates shared memory segment with key 2345, having size 1024 bytes. IPC\_CREAT is used to create the shared segment if it does not exist. 0666 are the permissions on the shared segment

printf("Key of shared memory is %d\n",shmid);

shared\_memory=shmat(shmid,NULL,0);

//process attached to shared memory segment

printf("Process attached at %p\n",shared\_memory);

//this prints the address where the segment is attached with this process

printf("Enter some data to write to shared memory\n");

read(0,buff,100); //get some input from user

strcpy(shared\_memory,buff); //data written to shared memory

printf("You wrote : %s\n",(char \*)shared\_memory);

}