Task 1

Create a private shared memory in C/C++. The process then creates a child and waits for the child to write the file’s contents to shared memory. The parent then reads the shared memory and changes the case of each character and removes all integers from the data. The child reads it back and writes the changed data back to the same file. (The file name is passed as command line argument).

Code:

|  |  |
| --- | --- |
| #include <stdio.h>  #include<string.h>  #include <sys/wait.h>  #include <sys/types.h>  #include<sys/ipc.h>  #include<sys/shm.h>  int main(int argc, char\* arg[])  {  key\_t key=1234 ;  pid\_t pid;  int shv=shmget(key, 1024, 0666|IPC\_CREAT|IPC\_EXCL);  pid=fork();  if(pid==0){  int shv=shmget(key, 1024, 0666);  char \*c = (char\*) shmat(shv,NULL,0);  int i=0;  char a,ar[100];  FILE \*fil;  fil=fopen(arg[1],"r");  do{  c[i]=fgetc(fil);  i++;  }while(c[i-1] != EOF);  shmdt(c);  fclose(fil);  }  else if (pid>0){  int shv=shmget(key, 1024, 0666);  char \*d = (char\*) shmat(shv,NULL,0);  int k,r; | wait(NULL);  for(k=0;k<strlen(d)-1;k++)  {  if(d[k]>='a'&&d[k]<='z')  d[k]=d[k]-32;  else if(d[k]>='A'&&d[k]<='Z')  d[k]+=32;  else if(d[k]>='0'&&d[k]<='9')  d[k]=' ';  else  d[k]=d[k];  }  pid\_t pid1;  pid1=fork();  if(pid1==0){  wait(NULL);  FILE \*f2;  f2=fopen(arg[1],"w");  for(k=0;k<strlen(d)-1;k++)  fputc(d[k],f2);  fclose(f2);  }  shmdt(d);  }  else  {  printf("Error \n\n");  }  shmctl(shv,IPC\_RMID,NULL);  return 0;  } |

Output:

Before:

****

**After:**

****

Task 2

Create a C++/C program that creates a shared memory and waits for the other process to write n data of n number of Students. The process that created the shared memory then writes the data of students to the file.

Code:

|  |  |
| --- | --- |
| #include <stdio.h>  #include<string.h>  #include <sys/wait.h>  #include <sys/types.h>  #include<sys/ipc.h>  #include<sys/shm.h>  struct student  {  char\* naam;  char\* hindsa;  }  int main(int argc, char\* arg[])  {  key\_t key ;  int shv=shmget(key, 1024, 0666|IPC\_CREAT|IPC\_EXCL);  pid\_t pid;  student \*s = (student \*) shmat(shv,(void\*)0,0);  pid=fork();  if(pid==0){  int a,n;  char\* A,\*B;  printf("Enter the number of students\n"); | scanf("%d",&a);  s=new student[a];  for(n=0;n<a;n++)  {  printf("Enter name and roll number");  scanf("%s %s",A,B);  s[n].naam=A;  s[n].hindsa=B;  }  }  else if (pid>0){  wait(NULL);  for  }  else  {  printf("Error \n\n");  }  shmdt(s);  shmctl(shv,IPC\_RMID,NULL);  return 0;  } |

Text, table

Description automatically generated with medium confidence