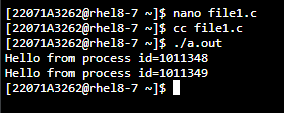
**Week – 1**

**1.Write a C program to demonstrate fork()**

**Code:**

#include<stdio.h>  
  
int main(){  
    fork();  
    printf("Hello from process id=%d\n",getpid());  
    return 0;  
}

**Output:**

****

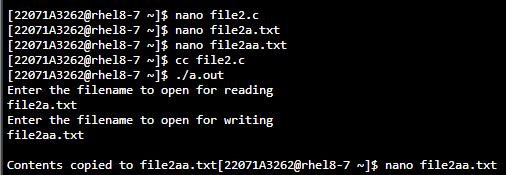
**2.Write a C program to copy content of one file to another**

**Code:**

#include <stdio.h>  
#include <stdlib.h>  
int main()  
{  
    FILE \*fptr1, \*fptr2;  
    char filename[100], c;  
    printf("Enter the filename to open for reading \n");  
    scanf("%s", filename);  
    fptr1 = fopen(filename, "r");  
    if (fptr1 == NULL){  
        printf("Cannot open file %s \n", filename);  
        exit(0);  
    }  
    printf("Enter the filename to open for writing \n");  
    scanf("%s", filename);  
    fptr2 = fopen(filename, "w");  
    if (fptr2 == NULL){  
        printf("Cannot open file %s \n", filename);  
        exit(0);  
    }  
    c = fgetc(fptr1);  
    while (c != EOF){  
        fputc(c, fptr2);  
        c = fgetc(fptr1);  
    }

printf("\nContents copied to %s", filename);  
    fclose(fptr1);  
    fclose(fptr2);  
    return 0;  
}

**Output:**

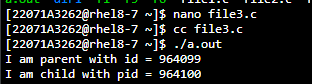
****

**3.Write a C program to demonstrate getpid()**

**Code:**

#include<stdio.h>  
#include<unistd.h>  
  
int main(){  
    int pid = fork();  
    if(pid > 0)  
        printf("I am parent with id = %d\n",getpid());  
    if(pid==0)  
        printf("I am child with pid = %d\n",getpid());  
   return 0;  
}

**Output:**

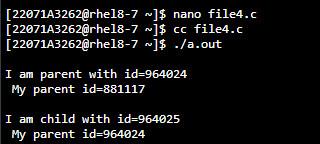
****

**4.Write a C program to demonstrate getppid()**

**Code:**

#include<stdio.h>  
  
int main(){  
    int pid = fork();  
    if(pid>0)  
        printf("I am parent with id=%d\n My parent id=%d",getpid(),getppid());  
    if(pid==0)  
        printf("I am child with id=%d\n My parent id=%d",getpid(),getppid());  
    return 0;  
}

**Output:**

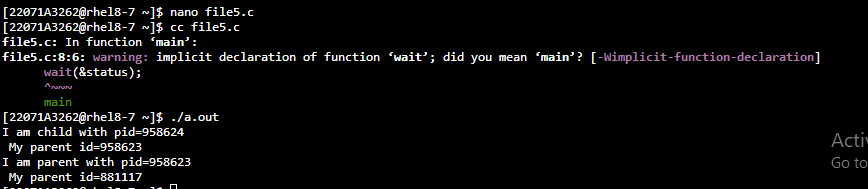
****

**5.Write a C program to demonstrate wait()**

**Code:**

#include<stdio.h>  
  
int main(){  
    int status;  
    int pid = fork();  
    if(pid>0){  
     wait(&status);  
     printf("I am parent with pid=%d\n My parent id=%d\n",getpid(),getppid());  
     }  
    if(pid==0)  
      printf("I am child with pid=%d\n My parent id=%d\n",getpid(),getppid());  
    return 0;    
}

**Output:**

****

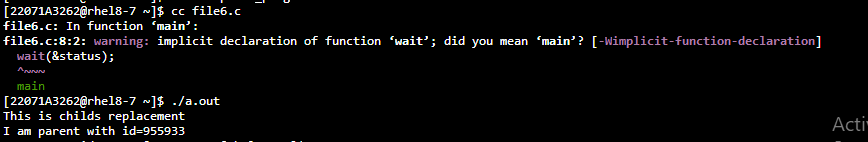
**6.Write a C program to demonstrate execve()**

**Code:**

#include<stdio.h>  
#include<unistd.h>  
  
int main(){  
    int status;  
    int pid = fork();  
    if(pid>0){  
        wait(&status);  
        printf("\nI am parent with id=%d\n My parent id=%d",getpid(),getppid());  
    }  
    if(pid==0)  
        execve("replace",NULL,NULL);  
    return 0;  
}

#include<stdio.h>  
int main(){  
   printf("This is childs replacement");  
   return 0;  
}

**Output:**

****