VIT®

Vellore Institute of Technology

(Deemed to be University under section 3 of UGC Act, 1956)

B. Tech. Semester 2020-2021

SCHOOL OF COMPUTER SCIENCE ENGINEERING

(SCOPE)

OPERATING SYSTEMS

FAT LAB

AMIT KUMAR

19BCE1281

First

Question

Write a C program to develop the user defined procedure call to check the existence of the given input string in the specific file. If the file contains the given input string, then extract the string and print it in a standard output device. Use the system call to do the same process. Then compare the overhead of user defined procedure call and system call.

Code

```
#include <string.h>
#include <stdlib.h>
#include <stdio.h>
#include <sys/time.h>
#include <unistd.h>
#include <assert.h>


void find()
{
        int num =0;
        char word[2000];
        char string[50];
        char student[] = "Amit";
                FILE *in_file = fopen("student.txt", "r");
                if (in_file == NULL)
                {
                        printf("Error file missing\n");
                        exit(-1);
                }
                while ( fscanf(in_file,"%s", string) == 1)
                {
                        //Add a for loop till strstr(string, student) does-
not returns null.
                        if(strstr(string, student)!=0) {//if match found
                                num++;
                        }
                }
                printf("we found the word %s in the file %d times\n",student,n
um );
                num = 0;
                fclose(in_file);
 }
```

```
 void find_using_grep(){
     FILE *cmd;
    char result[1024];
    FILE *popen(const char *command, const char *mode);
    int pclose(FILE *stream);


    cmd = popen("grep -i 'Amit' student.txt", "r");
    if (cmd == NULL) {
        perror("popen");
        exit(EXIT_FAILURE);
    }
    while (fgets(result, sizeof(result), cmd)) {
        printf("%s", result);
    }
    pclose(cmd);
}

long nanosec(struct timeval t){ /* Calculate nanoseconds in a timeval structur
e */
  return((t.tv_sec *1000000+t.tv_usec)*1000);
}

int main(){
  FILE* fileHandler;
  int i,j,res;
  long N_iterations=1; /* A million iterations */
  float avgTimeSysCall, avgTimeFuncCall;
  struct timeval t1, t2, t_test;

  /* Find average time for System call */
  res=gettimeofday(&t1,NULL); assert(res==0);
  for (i=0;i<N_iterations; i++){
    //fileHandler = fopen("test.txt", "w+");
    find_using_grep();
  }
  res=gettimeofday(&t2,NULL);   assert(res==0);
  avgTimeSysCall = (nanosec(t2) - nanosec(t1))/(N_iterations*1.0);

  /* Find average time for Function call */
  res=gettimeofday(&t1,NULL);  assert(res==0);
  for (i=0;i<N_iterations; i++){
    find();
  }
  res=gettimeofday(&t2,NULL);   assert(res==0);
  avgTimeFuncCall = (nanosec(t2) - nanosec(t1))/(N_iterations*1.0);

printf("The following result is for %ld iterations \n", N_iterations);
  printf("Average time for System call getpid : %f\n",avgTimeSysCall);
```

```
  printf("Average time for Function call : %f\n",avgTimeFuncCall);
  return 0;
}
```

Result

```
amit@amit-kumar-operat-VirtualBox: ~/fat/Codes
amit@amit-kumar-operat-VirtualBox:~/fat/Codes$ gcc first.c -o first
amit@amit-kumar-operat-VirtualBox:~/fat/Codes$ ./first
Amit
we found the word Amit in the file 1 times
The following result is for 1 iterations
Average time for System call getpid : 3866000.000000
Average time for Function call : 31000.000000
amit@amit-kumar-operat-VirtualBox:~/fat/Codes$
```

Student.txt

```
first.c 3    student.txt
student.txt
1   hello.
2   You will find my name at the end of the file
3   f
4   f
5   f
6   f
7   f
8   f
9   f
10  19bce1281
11  Amit
12  The above letter should be found.
13
```

Second

Question

Write a C program to create a new process. This newly created process reads the file access time for 5 different files from the main memory and plots the graph to show the relationship between file size and its access time from the main memory. User has to prompt the 5 different file names at runtime. The parent should wait until the child has to complete its task. Then, the parent has to execute the UNIX commands to print the system's current date (don't print the time) and number of files in the current directory.

Code

```
#include<stdio.h>
#include<sys/types.h>
#include<signal.h>
#include<string.h>
#include<time.h>
#include<unistd.h>
#include<sys/time.h>
#include<time.h>
#include <dirent.h>


  void access_child() {
    char data1[600], ch;
    int i, j, res, seq[15], ran[15], n;
    long int size[15];
    printf("enter the value of n to read last 'n' charecters");
    scanf("%d", &n);
    struct timeval t1, t2;
    FILE * f[5];
    //creating 15 text files files
    f[0] = fopen("1.txt", "r+");
    f[1] = fopen("2.txt", "r+");
    f[2] = fopen("3.txt", "r+");
    f[3] = fopen("4.txt", "r+");
    f[4] = fopen("5.txt", "r+");
    //finding file size
    for (i = 0; i < 5; i++) {
      fseek(f[i], 0L, SEEK_END);
      size[i] = ftell(f[i]);
      printf("size[%d]: %ld\n", i, size[i]);
    }
    //finding time for sequential access
```

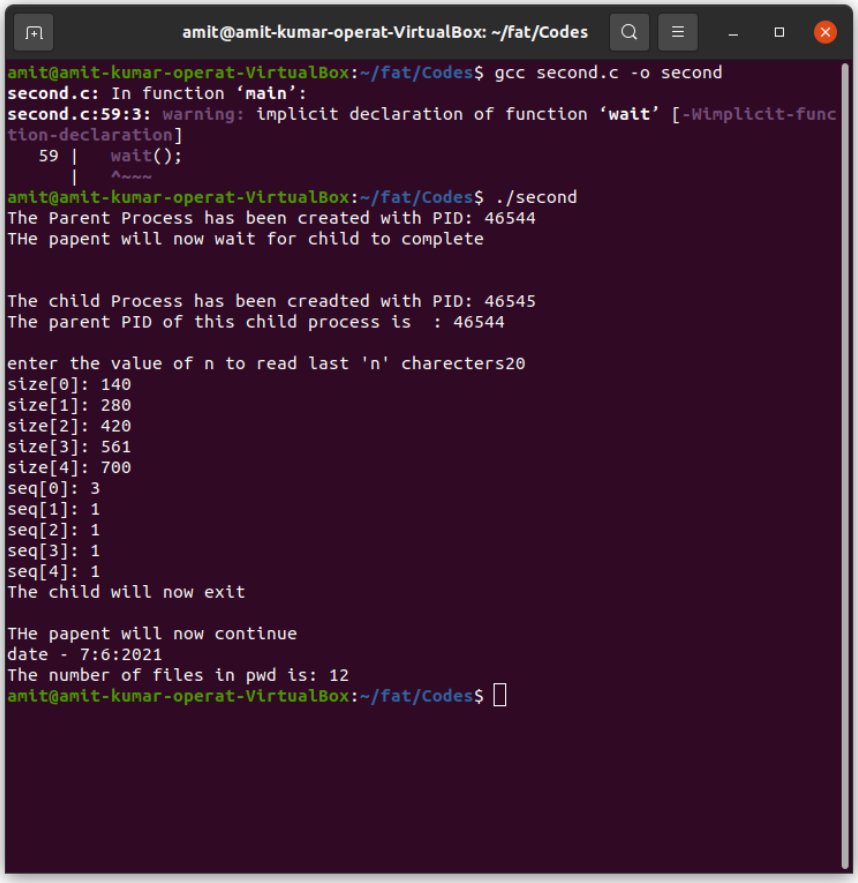
```
    for (i = 0; i < 5; i++) {
      gettimeofday( & t1, NULL);
      while (fgets(data1, 250, f[i]) != NULL) {}
      gettimeofday( & t2, NULL);
      long seconds = (t2.tv_sec - t1.tv_sec);
      seq[i] = (((seconds * 1000000) + t2.tv_usec) -
        (t1.tv_usec));
      printf("seq[%d]: %d\n", i, seq[i]);
    }
  }

int main(){
  int n;
  n = fork();
  if(n == 0){
    printf("\n");
    printf("The child Process has been creadted with PID: %d\n",getpid());
    printf("The parent PID of this child process is  : %d\n",getppid());
    printf("\n");
    access_child();
    printf("The child will now exit\n\n");
  }
  else{
    printf("The Parent Process has been created with PID: %d\n", getpid());
    printf("THe papent will now wait for child to complete\n");
    printf("\n");
    wait();
    time_t t = time(NULL);
    printf("THe papent will now continue\n");
    struct tm time = *localtime(&t);
               printf("date - %d:%d:%d\n", time.tm_mday,  time.tm_mon + 1, tim
e.tm_year + 1900);
               int file_count = 0;
    DIR * dirp;
    struct dirent * entry;

    dirp = opendir("/home/amit/fat/Codes");
    while ((entry = readdir(dirp)) != NULL) {
          if (entry->d_type == DT_REG) {
              file_count++;
          }
    }
    printf("The number of files in pwd is: %d\n", file_count);
    closedir(dirp);

  }
}
```

Result



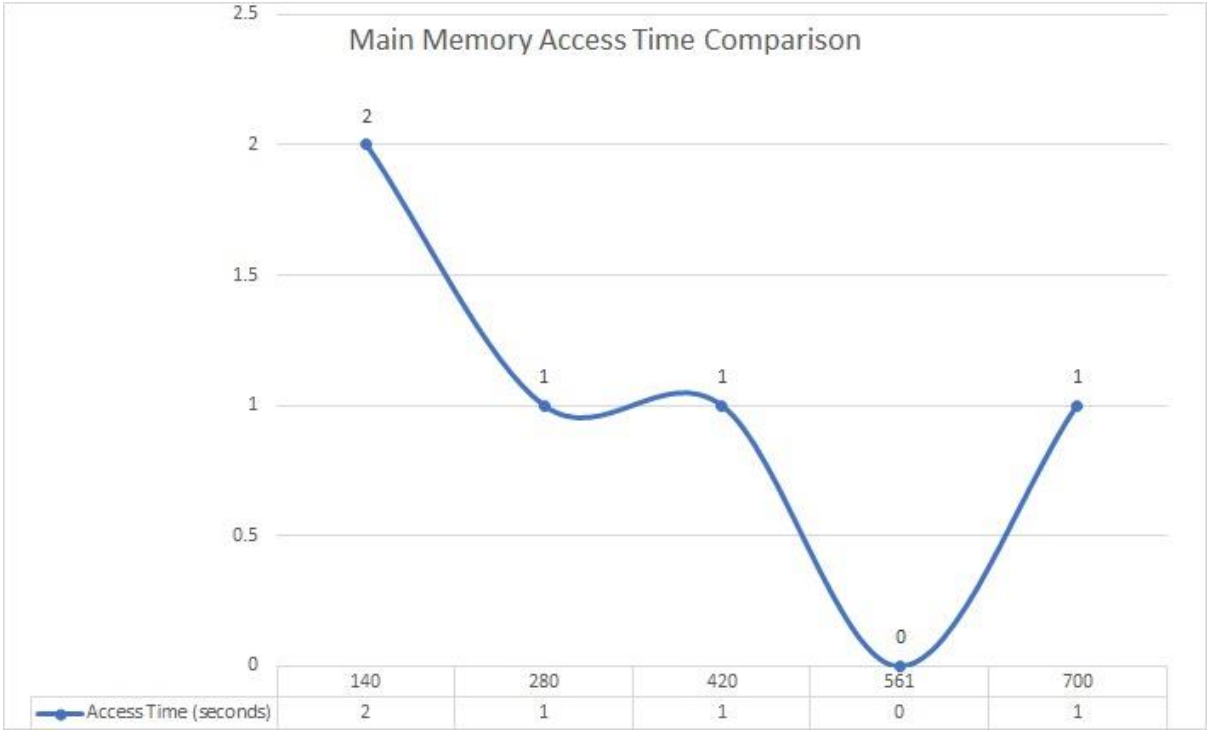
```
amit@amit-kumar-operat-VirtualBox: ~/fat/Codes
amit@amit-kumar-operat-VirtualBox:~/fat/Codes$ gcc second.c -o second
second.c: In function ‘main’:
second.c:59:3: warning: implicit declaration of function ‘wait’ [-Wimplicit-func
tion-declaration]
   59 |   wait();
      |   ^~~~
amit@amit-kumar-operat-VirtualBox:~/fat/Codes$ ./second
The Parent Process has been created with PID: 46544
THe papent will now wait for child to complete


The child Process has been creadted with PID: 46545
The parent PID of this child process is  : 46544

enter the value of n to read last 'n' charecters20
size[0]: 140
size[1]: 280
size[2]: 420
size[3]: 561
size[4]: 700
seq[0]: 3
seq[1]: 1
seq[2]: 1
seq[3]: 1
seq[4]: 1
The child will now exit

THe papent will now continue
date - 7:6:2021
The number of files in pwd is: 12
amit@amit-kumar-operat-VirtualBox:~/fat/Codes$
```

Graph


Main Memory Access Time Comparison
2.5
2
1.5
1
0.5
0
2
1
1
0
1
140
280
420
561
700
Access Time (seconds)
2
1
1
0
1