

DOKUMENTASI PRAKTIKUM

MODUL 2

SISTEM OPERASI (E)



Disusun oleh:

Setyassida Novian Putra D

(5114100024)

Jeffry Nasri Faruki

(5114100043)

Dosen Pembimbing :

Baskoro Adi Pratomo, S. Kom, M. Kom

Nomor 1

PEMBUATAN SHELL

1. Fungsi Main

```
int main(){
    char *input;
    char **args;
    char *pwd; // *char menunjuk pwd sekarang
    char buff[100]; //Buff pwd
    while(1){
        pwd=getcwd(buff, sizeof(buff)); // Mengecek pwd terkini
        printf(KBLU "%s#" KYEL "%s" KGRN " > $ " RESET, getenv("USER"), getenv("GDMSESSION"), pwd); //Template inputan user

        signal(SIGINT, signal_handler); //Untuk signal ctrl-c
        signal(SIGTSTP, signal_handler); //Untuk signal ctrl-z

        input=inputs();
        args=splitToToken(input);

        if(strcmp(args[0], "Exit")==0 || strcmp(args[0], "exit")==0){
            break;
        }

        if(!isBackgroundProcess(args)){
            process(args);
        }else{
            int index_apersand= isBackgroundProcess(args);
            args[index_apersand]=NULL;
            backgroundprocess(args);
        }

        free(input);
        free(args);
        return 0;
    }
}
```

2. Fungsi yang menerima Input User

```
char* inputs(){ // Fungsi untuk menerima inputan user
    int Max=MaxInput;
    char* input=malloc(sizeof(char)*Max);
    getline(&input, &Max, stdin);
    return input;
}
```

3. Fungsi yang memecah input user

```
void backgroundprocess(char**args){ //Fungsi membuat proses dr inputan user
    pid_t process,sid;
    int status;
    process=fork();
    if(process==0){
        return ;
    }else{
        umask(0);
        sid = setsid();
        if (strcmp(args[0], "cd")==0){
            chdir(args[1]); //arg[1] menunjukan direktori setelah user mengetik cd
        }else{
            execvp(args[0], args); // Eksekusi Inputan user
        }
    }
}
```

4. Fungsi yang mengeksekusi inputan yang telah dipecah

```
void process(char**args){ //Fungsi membuat proses dr inputan user
    pid_t process,wpid;
    int status;
    process=fork();
    if(process==0){
        if (strcmp(args[0], "cd")==0){
            chdir(args[1]); //arg[1] menunjukan direktori setelah user mengetik cd
        }else{
            execvp(args[0], args); // Eksekusi Inputan user
        }
    }else{
        do {
            wpid = waitpid(process, &status, WUNTRACED); // Menunggu proses anak
        } while (!WIFEXITED(status) && !WIFSIGNALED(status));
    }
}
```

5. Mengecek apakah inputan meminta proses background atau tidak

```
int isBackgroundProcess(char**args){
    int i=0;
    int Max=MaxInput;
    for(i=0;i<MaxInput;i++){
        if(args[i]==NULL){
            return 0;
        }else if(strcmp(args[i],"&")==0){
            return i;
        }
    }
}
```

6. Mengeksekusi inputan user secara background proses

```
char **splitToToken(char* input){ // Fungsi untuk memecah inputan user
    int Max=MaxInput;
    int index=0;
    char *token; // Inputan utuh
    char **tokens=malloc(sizeof(char)*Max); // tokens untuk menampung pecahan2 token
    token = strtok (input, "\n \t\r\a");
    while(token!=NULL){
        tokens[index]=token;
        index++;
        token=strtok(NULL, "\n \t\r\a");
    }
    return tokens;
}
```

7. Menangani sinyal ctrl-c dan ctrl-z

```
void signal_handler(int signo) // Fungsi penangkap sinyal
{
    if (signo == SIGTSTP){
        printf("\n");
        main();
        signal(SIGTSTP, SIG_IGN); //Untuk signal ctrl-z
    }
    else if (signo == SIGINT){
        printf("\n");
        main();
        signal(SIGINT, SIG_IGN); //Untuk signal ctrl-c
    }
}
```

8. Contoh penggunaan perintah ps

```
jeffry@nasri-HP-Pavilion-g4-Notebook-PC ~ $ gcc -o shell shell.c
jeffry@nasri-HP-Pavilion-g4-Notebook-PC ~ $ ./shell
jeffry@cinnamon/home/jeffry > $ ps
  PID TTY          TIME CMD
 3258 pts/1    00:00:00 bash
 4199 pts/1    00:00:00 shell
 4200 pts/1    00:00:00 ps
159371272 jeffry@cinnamon/home/jeffry > $
```

9. Contoh penggunaan perintah cd

```
jeffry#cinnamon/home/jeffry > $ cd Documents
jeffry#cinnamon/home/jeffry/Documents > $ ls
Asistensil    out.txt      signal.c     strtok       tugasignal
bash-3.2      Praktikum1 sinyalctrl-z.c strtok.c      tugassignal.c
Bersih2.psd  praktikum1.sh sinyalinterupt tambah.sh     Untitled-1.psd
bin          Praktikum2  sinyalinterupt.c test
fork         SEsilab1    strstr       thread
fork.c       signal      strstr.c     Threadcreating.c
jeffry#cinnamon/home/jeffry/Documents > $ exit
jeffry#cinnamon/home/jeffry > $ exit
jeffry@nasri-HP-Pavilion-g4-Notebook-PC ~ $
```

Nomor 2

PENGHITUNG JUMLAH BILANGAN PRIMA

1. Fungsi Main, dimana pembuatan thread dilakukan sebanyak jumlah input kita, dan tiap thread akan mengecek apakah bilangan tersebut prima atau bukan

```
void main () {
    int input,i=1,j=1,banyak_bilprim=0;
    printf("Input some number : ");scanf("%d",&input);
    pthread_t t1[input];
    void* bilprim;
    while(i <= input){
        pthread_create(&t1[i], NULL, find_prime, (void*)i);
        i++;
    }
    while(j <= input){
        pthread_join(t1[j], &bilprim);
        if(bilprim!=0){
            printf("%d ",(int)bilprim);
            banyak_bilprim++;
        }
        j++;
    }
    printf("\nTerdapat %d bilangan prima \n",banyak_bilprim);
}
```

2. Fungsi pengecekan bilangan prima, jika prima akan return ke nilai primanya. Jika BUKAN prima akan return 0

```
#include<stdio.h>
#include <stdlib.h>
#include <pthread.h>
void *find_prime(void *args) {
    int maks= (int)args;
    int count;
    if(maks==2){
        return (void *)maks;
    }
    for(count=2;count<maks;count++){
        if(maks%count==0){
            break;
        }else if(count==(maks-1)){
            return (void *)maks;
        }else{
            continue;
        }
    }
    pthread_exit(NULL); // Kalau tidak ada ini, jika maks-nya 2 maka return nya 2 kali
}
```

3. Contoh program dijalankan

```
jeffry@nasri-HP-Pavilion-g4-Notebook-PC ~ $ ./Find_Prime_and_Copy_File
Input some number : 8
2 3 5 7
Terdapat 4 bilangan prima
jeffry@nasri-HP-Pavilion-g4-Notebook-PC ~ $ ./Find_Prime_and_Copy_File
Input some number : 11
2 3 5 7 11
Terdapat 5 bilangan prima
jeffry@nasri-HP-Pavilion-g4-Notebook-PC ~ $
```

Nomor 3

Aplikasi Multithread untuk Menyalin Isi File

Berikut header file yang digunakan dalam source code.

```
1  #include <stdio.h>
2  #include <pthread.h>
3  #include <stdlib.h>
4
```

Func1 berfungsi untuk membaca dari file 1 dan menyalin isinya ke file 2.

```
5  void *func1() //fungsi untuk membaca dari file 1 dan menyalin isinya ke file 2
6  {
7      FILE *rf, *wf; //deklarasi tipe data file pointer
8      char c;
9      rf=fopen("file1.txt", "r"); //membuka file1.txt
10     wf=fopen("file2.txt", "w"); //membuka file2.txt
11     while((c=fgetc(rf))!=EOF) //looping untuk copy char dari file 2 ke file 3
12     {
13         fprintf(wf, "%c", c);
14         //fputc(c, wf);
15     }
16     fclose(rf); //menutup file1.txt
17     fclose(wf); //menutup file2.txt
18 }
```

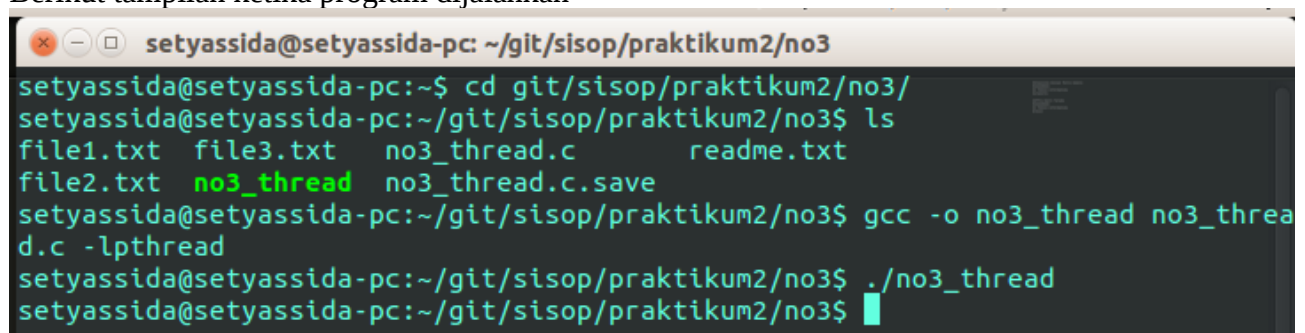
Func2 berfungsi untuk membaca dari file 2 dan menyalin isinya ke file 3.

```
20 void *func2() //fungsi untuk membaca dari file 2 dan menyalin isinya ke file 3
21 {
22     FILE *rf, *wf;
23     char c;
24     rf=fopen("file2.txt", "r"); //membuka file2.txt
25     wf=fopen("file3.txt", "w"); //membuka file3.txt
26     while((c=fgetc(rf))!=EOF) //looping untuk copy char dari file 2 ke file 3
27     {
28         fprintf(wf, "%c", c);
29         //fputc(c, wf);
30     }
31     fclose(rf); //menutup file2.txt
32     fclose(wf); //menutup file3.txt
33 }
```

Berikut **main function** dalam source code.

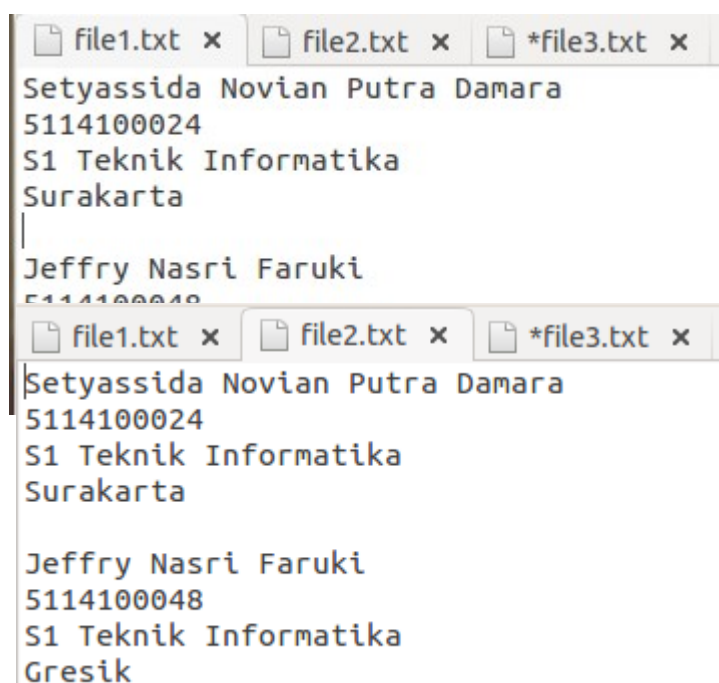
```
35 int main()
36 {
37     pthread_t t1, t2;
38     pthread_create(&t1, NULL, func1, NULL); //membuat thread 1 yang bertujuan menjalankan func1
39     pthread_create(&t2, NULL, func2, NULL); //membuat thread 2 yang bertujuan menjalankan func2
40     pthread_join(t1, NULL); //join thread 1
41     pthread_join(t2, NULL); //join thread 2
42     return 0;
43 }
```

Berikut tampilan ketika program dijalankan



```
setyassida@setyassida-pc: ~/git/sisop/praktikum2/no3
setyassida@setyassida-pc:~/git/sisop/praktikum2/no3$ cd git/sisop/praktikum2/no3/
setyassida@setyassida-pc:~/git/sisop/praktikum2/no3$ ls
file1.txt  file3.txt  no3_thread.c  readme.txt
file2.txt  no3_thread  no3_thread.c.save
setyassida@setyassida-pc:~/git/sisop/praktikum2/no3$ gcc -o no3_thread no3_threa
d.c -lpthread
setyassida@setyassida-pc:~/git/sisop/praktikum2/no3$ ./no3_thread
setyassida@setyassida-pc:~/git/sisop/praktikum2/no3$
```

Hasil output dari file 1, file 2, dan file 3



```
file1.txt x file2.txt x *file3.txt x
Setyassida Novian Putra Damara
5114100024
S1 Teknik Informatika
Surakarta
Jeffry Nasri Faruki
5114100048
file1.txt x file2.txt x *file3.txt x
Setyassida Novian Putra Damara
5114100024
S1 Teknik Informatika
Surakarta
Jeffry Nasri Faruki
5114100048
S1 Teknik Informatika
Gresik
```

file1.txt x file2.txt x file3.txt x

Setyassida Novian Putra Damara
5114100024
S1 Teknik Informatika
Surakarta

Jeffry Nasri Faruki
5114100048
S1 Teknik Informatika
Gresik