

Laporan

Nama : Muhammad Hanif (5114100013)

Ahmad Ismail H W (5114100032)

GIT

Langkah-langkah :

1. Register dulu pada github.com, isikan email, username, dan password. Lalu konfirmasi email anda.
2. Setelah itu, login pada github.com dengan username dan password anda, kemudian bikin repository baru
3. Install git pada linux anda dengan mengetik “sudo apt-get install git” pada terminal anda, jika tidak bisa, anda harus mengetik “sudo apt-get update” terlebih dahulu. Anda bisa mengecek versi git yang anda install dengan mengetikkan “git –version”.
4. Buat folder untuk di sambungkan dengan repository yang ada di github
5. Clone repository anda yang ada di github.com dengan “git clone <url clone pada repository github.com>” contoh : git clone www.github.com/ahmadismail10/Sisop.git
6. Apabila berhasil, anda sudah menyambungkan repository dengan folder yang anda buat.
7. Ada beberapa perintah tambahan untuk menggunakan git. Diantaranya :
 - Git add → untuk menambahkan file
 - Git commit → untuk menambahkan pesan pada file yang ditambahkan
 - Git push → untuk memasukkan file ke github.com
 - Git pull → untuk mendownload update yang terjadi pada github
8. Jika tidak bisa push file, lakukan “git config –global <user.name>” dan “git config –global <email>”

THREAD

- Deklarasi Thread

```
#include <pthread.h>
```

- Membuat Thread

```
Pthread_t nama variabel
```

```
Pthread_create(&nama variabel, NULL, nama fungsi thread, args)
```

- Compile program dengan Thread

```
Menggunakan -lpthread
```

```
Contoh penggunaan : gcc -o namafileexecutable namafile.c -lpthread
```

- Menunggu thread lain sebelum program selesai

pthread_join(namavariabel, NULL)

contoh :

```
testthread.c x
#include <stdio.h>
#include <pthread.h>

void *run (void *args)
{
    int i=0;
    for(i=1; i<=30; i++)
    {
        printf("thread 1 : %d\n", i);
    }
}

void *run2 (void *args)
{
    int i=0;
    for(i=1; i<30; i++)
    {
        printf("thread 2 : %d\n", i);
    }
}

void main()
{
    pthread_t t1, t2;
    pthread_create(&t1, NULL, run, NULL);
    pthread_create(&t2, NULL, run2, NULL);

    pthread_join(t2, NULL);

    int i;
    for (i=1; i<30; i++)
    {
        printf("Main : %d\n", i);
    }
}
```

Contoh hasilnya :

```
ahmad@ahmad-HP-Pavilion-15-Notebook-PC:~/Documents/Sistem Operasi$ ./testthread
thread 2 : 1
thread 2 : 2
thread 2 : 3
thread 2 : 4
thread 2 : 5
thread 2 : 6
thread 2 : 7
thread 2 : 8
thread 2 : 9
thread 2 : 10
thread 2 : 11
thread 2 : 12
thread 2 : 13
thread 2 : 14
thread 2 : 15
thread 2 : 16
thread 2 : 17
thread 2 : 18
thread 2 : 19
thread 2 : 20
thread 2 : 21
thread 2 : 22
thread 2 : 23
thread 1 : 1
thread 1 : 2
thread 1 : 3
thread 1 : 4
thread 1 : 5
thread 1 : 6
thread 1 : 7
thread 2 : 24
thread 2 : 25
thread 2 : 26
thread 2 : 27
thread 2 : 28
thread 2 : 29
thread 1 : 8
thread 1 : 9
thread 1 : 10
thread 1 : 11
thread 1 : 12
thread 1 : 13
thread 1 : 14
thread 1 : 15
thread 1 : 16
thread 1 : 17
thread 1 : 18
thread 1 : 19
thread 1 : 20
```

SIGNALING

Signal adalah sebuah event yang dikirimkan ke suatu proses oleh proses itu sendiri atau proses lain.

Kadang juga digunakan untuk mengirimkan notifikasi kepada suatu proses bahwa ada suatu event

- Definisi :

```
#include <sys/types.h>
```

```
#include <signal.h>
```

Untuk menangkap signal dibutuhkan signal handler kita bisa membuat signal handler dengan cara :

```
void signalhandler()
```

untuk memanggilnya bisa menggunakan system call signal (signal, namafungsi)

STRING

- Strstr

Strstr adalah sebuah fungsi yang digunakan untuk mencari sebuah lokasi substring yang dicari

deklarasi : strstr(string, index)

- Strtok

Strtok adalah fungsi yang digunakan untuk membagi string menjadi beberapa bagian yang dibatasi oleh karakter diinginkan.

Deklarasi : strtok(string, index)

jika selanjutnya variabel string yang digunakan sama, bisa di looping hingga strtok!=0 dan menggunakan strstr(NULL, index)

Contoh :

```
string.c x
#include <stdio.h>
#include <string.h>

int main ()
{
    char input[50];
    printf("Masukkan string : ");
    scanf("%[^\n]", input);
    printf("%s\n", input);

    char *substr = strstr(input, "o");
    printf("Hasil substr() : %s\n", substr);

    char *token = strtok(input, "o");
    printf("Hasil strtok() : %s\n", token);
    while(token!=NULL)
    {
        printf("Hasil strtok() : %s\n", token);
        token = strtok(NULL, "o");
    }
    return 0;
}
```

Hasil :

```
ahmad@ahmad-HP-Pavilion-15-Notebook-PC:~/Documents/Sistem Operasi$ ./string
Masukkan string : Hello World
Hello World
Hasil substr() : o World
Hasil strtok() : Hell
Hasil strtok() : Hell
Hasil strtok() :  World
ahmad@ahmad-HP-Pavilion-15-Notebook-PC:~/Documents/Sistem Operasi$
```

FORK dan EXEC

Fork ini digunakan untuk menduplikasi proses. Jadi membikin proses baru. Sehingga ada 2 proses yaitu file induk dan file anak. Sedangkan exec adalah system call untuk mengganti proses anak dari proses yang dibuat dari fork. Sehingga sekarang proses induk dan anak memiliki tugas atau menjalankan proses yang berbeda.

Latihan1.c x

```
#include<stdio.h>
#include<stdlib.h>
#include<unistd.h>

void Tugas1()
{
    int x,y,hasil;
    scanf("%d %d", &x, &y);
    hasil=x+y;
    printf("%d\n", hasil);
}

void Tugas2()
{
    execl("/bin/ls", "ls", "/home/ahmad", NULL);
}
void Tugas3()
{
    int i;
    for(i=1;i<=100;i++)
    {
        printf("%d\t", i);
    }
}

int main()
{
    pid_t pid;
    pid = fork();

    if(pid==0) Tugas1();
    else
    {
        wait(0);
        pid = fork();
        if(pid==0)
        {
            Tugas2();
        }
        else
        {
            wait(0);
            pid = fork();
            if(pid==0) Tugas3();
            else wait(0);
        }
    }
}
```

Contoh :

Hasil :

```
ahmad@ahmad-HP-Pavilion-15-Notebook-PC:~/Documents$ ./Latihan1
10
10
20
bash1.sh          hello.txt        sisop.txt
cekumur.sh        Latihan1        Templates
demo.sh           looping.sh       tesif.sh
Desktop           Music            Videos
Documents         Pictures         while.sh
Downloads         Public           xmind-linux-3.5.3.201506180105_amd64.deb
examples.desktop  sesilab1.sh

1      2      3      4      5      6      7      8      9      10     1
1      12     13     14     15     16     17     18     19     20     2
1      22     23     24     25     26     27     28     29     30     3
1      32     33     34     35     36     37     38     39     40     4
1      42     43     44     45     46     47     48     49     50     5
1      52     53     54     55     56     57     58     59     60     6
1      62     63     64     65     66     67     68     69     70     7
1      72     73     74     75     76     77     78     79     80     8
1      82     83     84     85     86     87     88     89     90     9
1      92     93     94     95     96     97     98     99     100    a
ahmad@ahmad-HP-Pavilion-15-Notebook-PC:~/Documents$
```