



Nama: **Bilhaq Avi Dewantara (120140141)**
Mata Kuliah: **Sistem Operasi (IF2223)**

Tugas Ke: **02**
Tanggal: 09/04/2022

1 Tujuan Hands On 2

Tujuan adanya Hands On kedua adalah untuk memahami bagaimana sistem bersinkronisasi dan permasalahan yang ada, serta juga memahami solusinya saat menjalankan critical section. Adapun beberapa implementasi yang diharuskan untuk dipahami pada Hands On kedua ini antara lain : *join* menggunakan semaphores, *Binary Semaphores*, *Produces Consumer*, *Reader/Writer Locks*, dan *Dining Philosophers*.

2 Fork/Join

2.1 Source Code

```
1  #include <stdio.h>
2  #include <stdlib.h>
3  #include <pthread.h>
4  #include <unistd.h>
5
6  #include "common.h"
7  #include "common_threads.h"
8
9  #ifdef linux
10 #include <semaphore.h>
11 #elif __APPLE__
12 #include "zemaphore.h"
13 #endif
14
15 sem_t s;
16
17 void *child(void *arg) {
18     sleep(2);
19     printf("child\n");
20     Sem_post(&s); // signal here: child is done
21     return NULL;
22 }
23
24 int main(int argc, char *argv[]) {
25     Sem_init(&s, 0);
26     printf("parent: begin\n");
27     pthread_t c;
28     Pthread_create(&c, NULL, child, NULL);
29     Sem_wait(&s); // wait here for child
30     printf("parent: end\n");
31     return 0;
32 }
33
```

2.2 Output

3 Pembahasan Tut 1

3.1 Tut 1.1 *echo*

Pada percobaan tut 1.1 ini akan mencoba perintah *echo* yang akan menampilkan pesan yang kita tuliskan. Pesan yang akan di tulis ialah "*Hello World*". Dan dengan begitu adanya *echo Hello World*, maka akan mengeluarkan *Hello World* di dalam terminal Linux.

3.2 Tut 1.2 *man*

Pada percobaan tut 1.2 ini akan mencoba perintah *man*, dengan mengetik perintah *man echo* di terminal, maka akan mengeluarkan fungsi-fungsi dari sebuah *command man* yang berguna dalam menampilkan sebuah fungsi. Dan dengan adanya *man echo*, maka akan menampilkan berbagai perintah dan kegunaan dari perintah *echo* yang dijelaskan secara mendetail.

3.3 Tut 1.3 *echo SHELL*

Pada percobaan tut 1.3 ini akan mencoba perintah *echo SHELL* yang akan menampilkan pesan yang kita tuliskan. Pesan yang akan di tampilkan dari perintah tersebut ialah mengeluarkan directory */bin/bash* pada layar terminal.

3.4 Tut 1.4 *who, cd, mkdir, touch, ls, cp, rm*

Pada percobaan tut 1.4 ini akan mencoba perintah *who, cd, mkdir, touch, ls, cp, rm* yang akan menampilkan informasi tentang *user, directory*, membuat *directory*, membuat file, menampilkan file, menyalin file, dan menghapus file.

4 Pembahasan Tut 2

4.1 Tut 2.1 *sed*

Pada percobaan tut 2.1 ini akan mencoba perintah *sed* yang diminta untuk menghapus 1 karakter di depan dan dibelakang di setiap baris code. Sebelumnya saya mencoba untuk membuat file *testing.txt* yang digunakan sebagai tempat eksekusi *sed*. Kemudian, barulah saya membuka terminal untuk mulai proses menghapus karakternya dengan perintah *sed* seperti di gambar.

4.2 Tut 2.2 *grep*

Pada percobaan tut 2.2 ini akan mencoba perintah *grep* yang diminta untuk mencari karakter tertentu pada sebuah file. Sebelumnya saya membuat file *testing.txt* yang digunakan sebagai tempat eksekusi *grep*. Kemudian, barulah saya membuka terminal untuk mulai proses pencarian kata *saya* pada file *testing.txt* seperti yang digambar.

5 Pembahasan Tut 3

5.1 Tut 3.1 *Shell Scripting*

Pada percobaan tut 3.1 ini akan mencoba perintah *nano* untuk membuat file *test.sh* yang digunakan sebagai tempat mengisi kata *echo Hello World*. Selanjutnya, untuk me-run file tersebut diperlukan mengetik *chmod +x test.SH* di terminal. Barulah selanjutnya saya mengetik *./test.sh* di terminal untuk menjalankannya.

6 Assingment 6

Source Code

```
1  #get filename
2  echo -n "Nama file : "
3  read bilhaqavidewantara
4
5  if [!-f $bilhaqavidewantara]
6  then
7  echo "Nama File $bilhaqavidewantara dosen not exist"
8  exit 1
9  fi
10 tr '[a-z]' '[A-Z]' < $bilhaqavidewantara
11
```

Penjelasan

Pada percobaan *Assignment 6* ini akan mencoba perintah untuk mengubah kalimat yang awalnya berhuruf kecil menjadi kalimat berhuruf besar semua. Pertama saya membuat file *Handson1_6_120140141.sh* berisi kode seperti gambar. Setelah itu, diperlukan membuat file *test_6.sh* yang berisi string bebas, dalam hal ini saya isi file tersebut dengan *"bilhaq avi dewantara"*. Kemudian, barulah membuka terminal untuk mengeksekusi program tersebut. Pastikan kedua file sudah berada di *directory* yang sama agar memudahkan.

7 Assignment 8

Source Code

```
1 echo "Input nama file : "  
2 read fname  
3 echo "Input line pertama yang ingin di output : "  
4 read s  
5 echo "Input line terakhir yang ingin di output : "  
6 read n  
7 sed -n $s,$n\p $fname | cat > newline  
8 cat newline
```

Penjelasan

Pada percobaan *Assignment 8* ini akan mencoba untuk menampilkan baris kalimat yang ingin di *output* dari file. Pertama saya membuat file *Handson1_8_120140141.sh* berisi kode seperti gambar. Setelah itu, diperlukan membuat file *test_8.sh* untuk membuat kata-kata sebagai *input*. Untuk menjalankannya kita perlu membuka terminal dan user diminta untuk memasukkan baris awal yang akan di *output* dan baris akhir yang akan di *output* pada layar terminal.

8 Assignment 9

Source Code

```
1  echo "Masukkan kata untuk mencocokkan isi dalam file : "  
2  read pat  
3  for file in $@  
4  do  
5      if ! [ -r $file ]  
6      then  
7          echo "File tidak ada atau tidak terbaca!"  
8          continue  
9      fi  
10     echo "Sebelum -----"  
11     cat $file  
12     sed -i "$pat/d" $file  
13     echo "Sesudah -----"  
14     cat $file  
15 done
```

Penjelasan

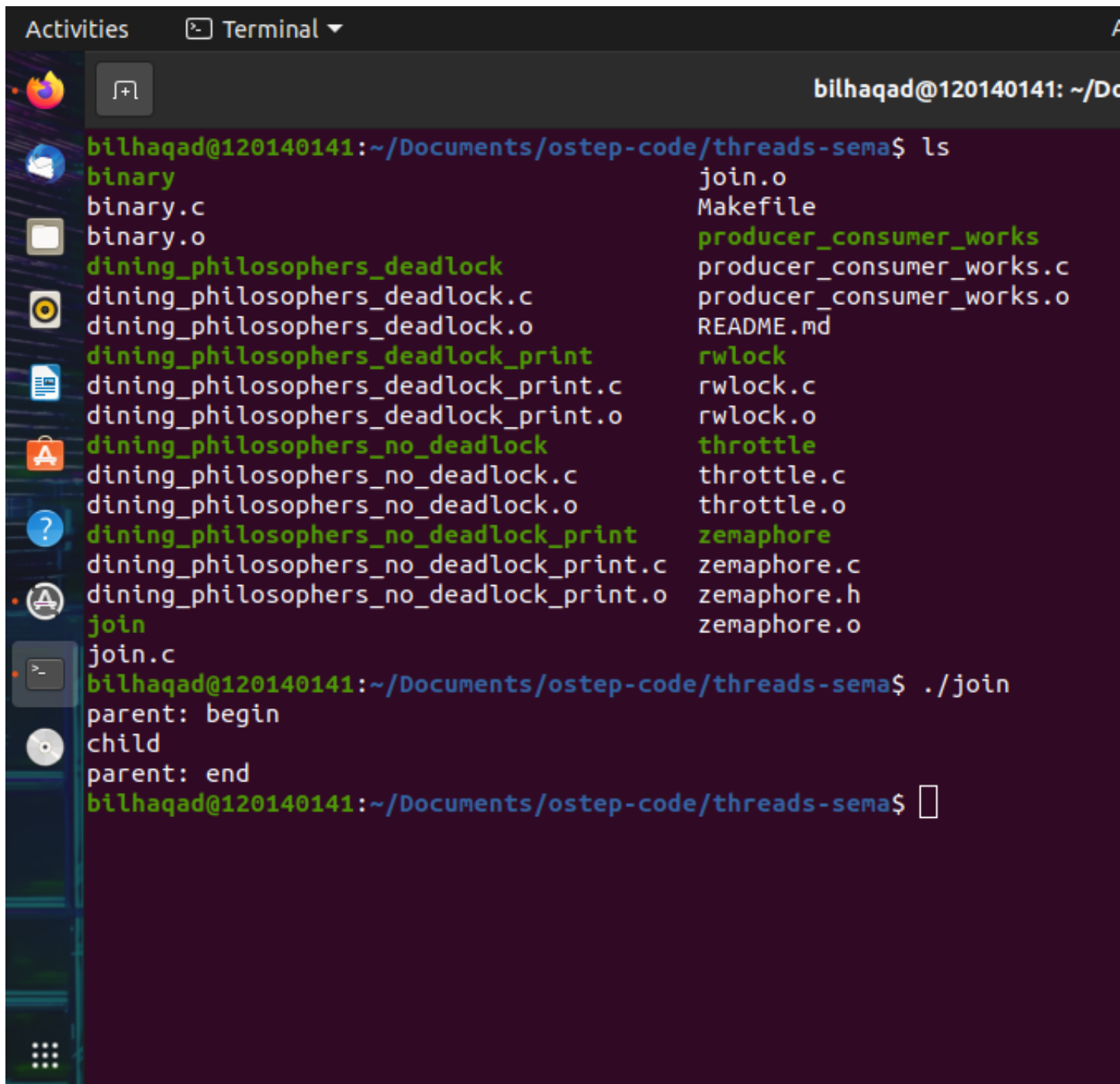
Pada percobaan *Assignment 9* ini akan mencoba untuk menghapus baris kalimat yang mengandung kata yang dicari. Pertama saya membuat file *Handson1_9_120140141.sh* berisi kode seperti gambar. Setelah itu, diperlukan membuat file *test_9.sh* untuk membuat kata-kata sebagai *input*. Untuk menjalankannya kita perlu membuka terminal dan *user* diminta untuk memasukkan kata untuk mencocokkan baris yang ingin di hapus dan akan di *output* pada layar terminal. Pada kode kali ini menggunakan implementasi perintah *echo*, *read*, *for*, *do*, *if*, *then*, *continue*, *fi*, *cat*, dan *sed*.

9 Kesimpulan

Pada Hands On 1 ini yang saya dapatkan setelah menjalankannya ialah saya dapat mengenal sistem operasi linux khususnya Ubuntu 20.04 LTS ini meskipun hanya menggunakan Oracle VirtualBox. Selain itu, saya dapat mengetahui banyak hal dari tugas ini yaitu itu menggunakan konsep-konsep baru dan mengimplementasikannya pada terminal di linux. Dengan begitu, saya dapat menyelesaikan tugas ini sesuai dengan kemampuan yang saya miliki terutama menggunakan latex ini yang baru bagi saya, sehingga banyak sekali yang saya dapatkan dari tugas ini.

10 Link GitHub

Link GitHub dari Hands On 1 ini : [Klik disini](#)



The screenshot shows a terminal window titled "Terminal" with the user "bilhaqad@120140141" in the directory "~/Documents/ostep-code/threads-sema". The user has run the command `ls`, which lists the following files and directories:

- `binary`
- `binary.c`
- `binary.o`
- `dining_philosophers_deadlock`
- `dining_philosophers_deadlock.c`
- `dining_philosophers_deadlock.o`
- `dining_philosophers_deadlock_print`
- `dining_philosophers_deadlock_print.c`
- `dining_philosophers_deadlock_print.o`
- `dining_philosophers_no_deadlock`
- `dining_philosophers_no_deadlock.c`
- `dining_philosophers_no_deadlock.o`
- `dining_philosophers_no_deadlock_print`
- `dining_philosophers_no_deadlock_print.c`
- `dining_philosophers_no_deadlock_print.o`
- `join`
- `join.c`
- `join.o`
- `Makefile`
- `producer_consumer_works`
- `producer_consumer_works.c`
- `producer_consumer_works.o`
- `README.md`
- `rwlock`
- `rwlock.c`
- `rwlock.o`
- `throttle`
- `throttle.c`
- `throttle.o`
- `zemaphore`
- `zemaphore.c`
- `zemaphore.h`
- `zemaphore.o`

After listing the files, the user has run the command `./join`, which produces the following output:

```
parent: begin
child
parent: end
```

The terminal prompt is now `bilhaqad@120140141:~/Documents/ostep-code/threads-sema$` with a cursor.

Gambar 1: Fork/Join