

**UJIAN TENGAH SEMESTER GENAP TAHUN AKADEMIK 2024/2025**  
**PROGRAM STUDI INFORMATIKA**  
**FAKULTAS TEKNIK DAN INFORMATIKA**  
**UNIVERSITAS MULTIMEDIA NUSANTARA**

Mata Kuliah : IF 260L – Operating System (Praktikum)	Hari/Tanggal :
Dosen : A. A. N. Ananda Kusuma I Putu Ananta Yogiswara	Waktu :
Sifat Ujian : Open Source	Tipe Soal : Take Home

**KETENTUAN / PETUNJUK UJIAN:**

**INSTRUCTIONS**

- Pastikan kode anda rapi, mudah dibaca, dan dilengkapi dengan komentar yang memadai untuk menjelaskan logika pemrograman anda  
*Make sure your code is neat, easy to read, and supplemented by adequate comments to explain your programming logic.*
- File yang dikumpulkan adalah dalam bentuk archive dengan format tar.gz atau zip yang berisikan file-file .c dan hasil kompilasi. Penamaan file adalah sebagai berikut: **UTS-IF260L-NIM-NAMA.tar.gz**  
*The submitted file is in the form of an archive with tar.gz or zip format containing .c files and compilation results. The file naming is as follows: **UTS-IF260L-NIM-NAMA.tar.gz***
- Segala bentuk kecurangan** akan mengakibatkan **nilai 0**.  
***Any form of cheating** will result in a score of **ZERO***

**SUB-CAPAIAN PEMBELAJARAN MATA KULIAH (SUB-CPMK):**

SUB-CAPAIAN PEMBELAJARAN MATA KULIAH (SUB-CPMK)		CPL
Kode Sub-CPMK	Penjelasan Sub-CPMK	
Sub-CLO0211	Students are able to explain the concepts of operating system, operating-systems structures, and their relationship to computer systems	ELO 02
Sub-CLO0711	Students are able to explain the concepts of Process Management which consists of Process and Thread, Process Synchronization, CPU Scheduling, and Deadlock, and apply them in solving various system problems using a programming language.	ELO 07

**Soal:**

**QUESTIONS**

**1. (30 points; Sub-CLO0211)**

Buatlah program dengan menggunakan system call **fork()**, sehingga terbentuk hirarki process seperti di bawah. Process A menunggu string dari keyboard. Pengguna mengetik kalimat sebagai berikut: **“Ujian Praktikum Sistem Operasi”**. Process C memiliki procedure untuk mengubah string menjadi huruf besar semua. Dari hirarki process yang terbentuk dan string berisikan masukan dari pengguna, tampilkan ke layar monitor keluaran dari tiap process dengan urutan sbb.:

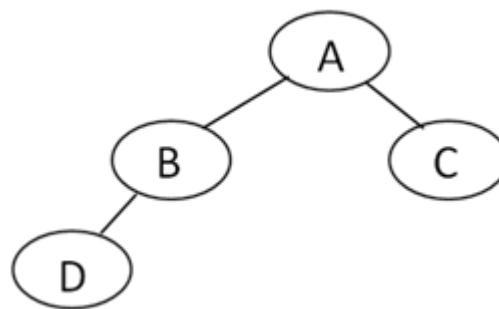
Create a program using the **fork()** system call, so that a process hierarchy is formed as below. Process A waits for a string from the keyboard. The user types the following sentence: “**Ujian Praktikum Sistem Operasi**”. Process C has a procedure to change the string to all uppercase. From the process hierarchy that is formed and the string containing input from the user, display the output of each process on the monitor screen in the following order:

“Process A: #PID #Parent\_PID : Ujian Praktikum Sistem Operasi”

“Process B: #PID #Parent\_PID : Ujian Praktikum Sistem Operasi”

“Process D: #PID #Parent\_PID : Ujian Praktikum Sistem Operasi”

“Process C: #PID #Parent\_PID : UJIAN PRAKTIKUM SISTEM OPERASI”



#### Rubrik Penilaian

Kriteria	Nilai	Max
Source code yang dikumpulkan dapat dikompilasi	5	30
Executable file yang dihasilkan dapat dijalankan	5	
Memahami penggunaan system call yang diminta untuk memperoleh hirarki process yang diminta	10	
Memahami mekanisme koordinasi antar process untuk menghindari terjadinya zombie process	10	
<b>Total</b>		<b>30</b>

## 2. (35 points; Sub-CLO0711)

Di sebuah restoran, disiapkan program-program untuk menerima pesanan yang kemudian dibaca oleh koki di dapur untuk dimasak. Untuk sinkronisasi proses bisnis antara pemesanan dan penyiapan makanan, pihak manajemen menyiapkan batasan maksimum jumlah pencatatan, yaitu 30.

Buatlah dua buah program **order.c** dan **cook.c** yang memodelkan mekanisme di atas. Kedua program tersebut dijalankan pada dua terminal terpisah, dan masing-masing menampilkan jumlah pemesanan, jumlah pemesanan pada catatan manajemen, dan jumlah pemesanan yang telah dimasak. Untuk memodelkan variasi waktu dalam melakukan pemesanan dan memasak, gunakan **sleep()** dengan argument waktu yang acak. Gunakan **shared memory** dan **semaphore** untuk komunikasi dan sinkronisasi kedua program tersebut.

In a restaurant, programs are prepared to receive orders which are then read by the chef in the kitchen to be cooked. To synchronize the business process between ordering and food preparation, the management sets a maximum limit on the number of records, which is 30.

Create two programs **order.c** and **cook.c** that model the above mechanism. Both programs are run on two separate terminals, and each displays the number of orders, the number of orders in the management record, and the number of orders that have been cooked. To model the variation in time in placing orders and cooking, use `sleep()` with a random time argument. Use **shared memory** and **semaphores** for communication and synchronization of the two programs.

#### Rubrik Penilaian

Kriteria	Nilai	Max
Source code yang dikumpulkan dapat dikompilasi	5	35
Executable file yang dihasilkan dapat dijalankan	5	
Memahami cara menggunakan system call yang diperlukan	5	
Memahami cara membuat segment shared memory	5	
Memahami cara koordinasi antara pencatatan pemesanan (producer) dan pengambilan data pemesanan (consumer)	5	
Memahami cara untuk melakukan sinkronisasi	10	
<b>Total</b>		<b>35</b>

### 3. (35 points; Sub-CLO0711)

Buatlah sebuah **multi-threaded** program yang memodelkan perlombaan minum kopi dengan spesifikasi sebagai berikut:

- Program ini memiliki 5 buah thread
- Tiga buah thread digunakan untuk tiga orang kontestan yang akan saling berlomba untuk minum, satu buah thread digunakan untuk menampilkan status lomba, dan satu lagi untuk thread dari main program.
- Program ini akan menerima input berupa integer jumlah cangkir kopi yang akan dilombakan, yaitu  $10 < n < 200$ . Ketiga thread akan berlomba menghabiskan paling banyak cangkir kopi.
- Perlombaan ini harus diatur sedemikian rupa agar adil dimana tidak terdapat thread yang memonopoli jalannya perlombaan.
- Berikan informasi yang jelas mengenai jalannya proses perlombaan dari awal dimulai hingga akhir selesai, misal dengan menampilkan perubahan ketersediaan cangkir dan jumlah cangkir yang telah digunakan oleh kontestan.
- Di akhir eksekusi, program akan menampilkan juara pertama hingga ketiga dari perlombaan ini.



Create a multi-threaded program that models a coffee drinking competition with the following specifications:

- This program has 5 threads
- Three threads are used for three contestants who will compete to drink, one thread is used to display the competition status, and one for the main program's thread.

- This program will receive input in the form of an integer number of cups of coffee to be competed, namely  $10 < n < 200$ . The three threads will compete to finish the most cups of coffee.
- This competition must be arranged in such a way that it is fair where no thread monopolizes the course of the competition.
- Provide clear information about the course of the competition from the beginning to the end, for example by displaying changes in the availability of cups and the number of cups that have been used by contestants.
- At the end of the execution, the program will display the first to third place winners of this competition.

#### Rubrik Penilaian

Kriteria	Nilai	Max
Source code yang dikumpulkan dapat dikompilasi	5	35
Executable file yang dihasilkan dapat dijalankan	5	
Memahami cara membuat tambahan thread	5	
Memahami cara melakukan komunikasi antar thread	10	
Memahami teknik sinkronisasi untuk menghindari masalah race condition	10	
<b>Total</b>		<b>35</b>

Acuan Pembuatan Soal:	Soal dibuat oleh:	Disetujui oleh:
Materi Praktikum Pertemuan 1 - 7	a.n Tim Dosen Mata Kuliah  (A. A. N. Ananda Kusuma, MEng, PhD) Koord. Mata Kuliah	 (Arya Wicaksana, SKom, MSc) Ketua Program Studi