

**UJIAN TENGAH SEMESTER GENAP TAHUN AKADEMIK 2023/2024**  
**PROGRAM STUDI INFORMATIKA**  
**FAKULTAS TEKNIK DAN INFORMATIKA**  
**UNIVERSITAS MULTIMEDIA NUSANTARA**

<b>Mata Kuliah</b> :	<b>IF260 – Operating System</b>	<b>Hari/Tanggal</b> :	
<b>Dosen</b> :	<b>Suwito Pomalingo A.A.N Ananda Kusuma Eka Jaya Harsono Muhammad Fahrury Romdendine Arko Djajadi</b>	<b>Jam</b> :	
<b>Sifat Ujian</b> :	<b>Open Resources</b>	<b>Tipe Soal</b> :	<b>Onsite ELearning Pakai Laptop Sendiri</b>

**KETENTUAN / PETUNJUK UJIAN:**

1. Pastikan kode Anda rapi, mudah dibaca, dan dilengkapi dengan komentar yang memadai untuk menjelaskan logika pemrograman Anda.
2. File yang dikumpulkan adalah file .c dan file yang telah dikompilasi, yang kemudian kedua file tersebut dikompres menggunakan zip atau kompresi lainnya seperti gz atau tar.gz dengan penamaan file UTS-IF260-NIM-NAMA LENGKAP.zip
3. Pastikan dalam file yang terkompresi terdapat 2 (dua) file di dalamnya.
4. Segala bentuk kecurangan yang terindikasi, maka akan mendapat nilai F / tidak lulus

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

<b>SUB-CAPAIAN PEMBELAJARAN MATA KULIAH (SUB-CPMK)</b>		<b>CPL</b>
<b>Kode Sub-CPMK</b>	<b>Penjelasan Sub-CPMK</b>	
Sub-CLO0211	Students are able to explain the concept of operating systems, operating-systems structures, and their relationship to computer systems	ELO 2
Sub-CLO0711	Students are able to explain the concept of Process Management which consists of Process and Thread, Process Synchronization, CPU Scheduling, and Deadlock, and apply it in the form of a programming language	ELO 7

**SOAL NOMOR 1 Sub-CLO0211, Bobot (10%)**

***Konteks***

System Call merupakan antarmuka pemrograman tingkat rendah antara ruang pengguna dan kernel sistem operasi. System Call memungkinkan program di ruang pengguna untuk meminta layanan dari kernel sistem operasi, seperti operasi file, pembuatan proses, dan komunikasi antar-proses.

***Studi Kasus***

Anda diminta untuk mengembangkan sebuah program dalam bahasa C yang menampilkan daftar file dalam sebuah direktori. Program ini harus menggunakan system call **opendir()**, **readdir()**, dan **closedir()** untuk membaca isi direktori. Direktori yang akan dibaca harus dapat ditentukan oleh pengguna melalui argumen baris perintah. Program juga harus mengecek apakah argumen yang diberikan merupakan direktori yang valid. Jika tidak, program harus menampilkan pesan kesalahan yang sesuai.

***Persyaratan***

1. Program harus dibuat dalam bahasa C.
2. Gunakan system call **opendir()**, **readdir()**, dan **closedir()** untuk membaca dan menampilkan isi dari direktori yang ditentukan.
3. Jika direktori tidak ditemukan atau argumen yang diberikan bukan direktori, tampilkan pesan kesalahan.
4. Jika direktori valid, tampilkan daftar nama file dan sub-direktori yang terdapat di dalamnya.
5. Pastikan program Anda dapat menangani lebih dari satu argumen direktori.

***Pertanyaan***

Tulislah kode dalam bahasa C yang memenuhi persyaratan di atas. Pastikan untuk menangani kemungkinan error yang mungkin terjadi selama eksekusi system call.

**UJIAN TENGAH SEMESTER GENAP TAHUN AKADEMIK 2023/2024**  
**PROGRAM STUDI INFORMATIKA**  
**FAKULTAS TEKNIK DAN INFORMATIKA**  
**UNIVERSITAS MULTIMEDIA NUSANTARA**

---

**Catatan**

Dalam menyelesaikan soal ini, Anda diharapkan untuk dapat memahami dan menerapkan konsep system call dalam konteks operasi file dan direktori. Program harus dapat dijalankan dengan benar dan menangani kondisi-kondisi khusus seperti input yang tidak valid atau adanya error saat eksekusi system call. Gunakan komentar dalam kode Anda untuk menjelaskan logika dan alur pemrograman yang Anda gunakan.

**Rubrik Penilaian**

Aspek/Kriteria yang Dinilai	Kriteria Penilaian				
	Sangat Kurang	Kurang	Cukup	Baik	Sangat Baik
	1	3	5	8	10
Source code	Tidak ada yang dikumpulkan/ kode terindikasi plagiarisme.	Ada kode yang dikumpulkan, tetapi <i>corrupt</i> , tidak dapat dikompilasi, atau tidak dapat dieksekusi.	Dapat dikompilasi dan dieksekusi, tetapi hasilnya salah semua.	Dapat dikompilasi dan dieksekusi, program menghasilkan keluaran yang tidak sepenuhnya salah.	Dapat dikompilasi, dieksekusi dan keluaran yang dihasilkan sesuai.

**SOAL NOMOR 2 Sub-CLO0711, Bobot (10%)****Konteks**

Dalam sebuah sistem operasi, proses sinkronisasi sangat penting untuk menghindari kondisi race dan memastikan akses sumber daya berbagi dilakukan secara teratur. Semaphore adalah mekanisme sinkronisasi yang memungkinkan proses untuk mengontrol akses terhadap sumber daya bersama melalui penggunaan counter. Semaphore biner, atau mutex, digunakan untuk kasus di mana hanya satu proses yang diizinkan mengakses sumber daya pada satu waktu.

**Studi Kasus**

Anda diminta untuk mengembangkan sebuah program dalam bahasa C yang mensimulasikan operasi pada printer yang digunakan bersama dalam sebuah lab komputer. Terdapat 5 proses yang ingin mencetak dokumen mereka pada printer tersebut. Untuk menghindari kondisi race dan memastikan bahwa setiap proses dapat mencetak dokumennya tanpa gangguan dari proses lain, Anda harus menggunakan semaphore.

**Persyaratan**

1. Buatlah struktur data untuk menyimpan informasi proses (misalnya, id proses dan dokumen yang akan dicetak).
2. Implementasikan semaphore untuk mengontrol akses ke printer.
3. Setiap proses harus menunggu giliran untuk mencetak dokumen mereka. Setelah selesai, proses tersebut harus melepaskan akses printer agar proses berikutnya dapat mencetak.
4. Tampilkan pesan pada konsol setiap kali proses memulai dan menyelesaikan pencetakan, termasuk id proses.

**Pertanyaan**

Tuliskan kode dalam bahasa C yang memenuhi persyaratan di atas. Gunakan semaphore untuk sinkronisasi proses. Anda bisa menggunakan fungsi-fungsi dari library `<semaphore.h>` dan `<pthread.h>` untuk manajemen thread dan semaphore.

**Catatan**

Dalam menyelesaikan soal ini, anda diharapkan memahami konsep semaphore dan bagaimana penerapannya dalam sinkronisasi proses di sistem operasi. Pastikan kode Anda rapi, mudah dibaca, dan dilengkapi dengan komentar yang memadai untuk menjelaskan logika pemrograman Anda.

**Rubrik Penilaian**

Aspek/Kriteria yang Dinilai	Kriteria Penilaian				
	Sangat Kurang	Kurang	Cukup	Baik	Sangat Baik
	1	3	5	8	10
<i>Source code</i>	Tidak ada yang dikumpulkan/ kode terindikasi plagiarisme.	Ada kode yang dikumpulkan, tetapi <i>corrupt</i> , tidak dapat dikompilasi, atau tidak dapat dieksekusi.	Dapat dikompilasi dan dieksekusi, tetapi hasilnya salah semua.	Dapat dikompilasi dan dieksekusi, program menghasilkan keluaran yang tidak sepenuhnya salah.	Dapat dikompilasi, dieksekusi dan keluaran yang dihasilkan sesuai.

**SOAL NOMOR 3 Sub-CLO0711, Bobot (10%)**

**Konteks**

Dalam manajemen proses pada sistem operasi, proses dapat berakhir namun masih mempertahankan informasi status di sistem sampai parent processnya membaca status tersebut, disebut sebagai proses Zombie. Sementara itu, proses Orphan adalah proses yang berjalan setelah parent processnya berakhir, biasanya diadopsi oleh init process (process dengan PID 1).

**Studi Kasus**

Anda diminta untuk mengembangkan sebuah program dalam bahasa C yang secara eksplisit menciptakan kondisi proses Zombie dan Orphan. Program harus terdiri dari parent process yang menciptakan child process. Child process tersebut kemudian menciptakan sub-child process.

**Persyaratan**

1. Parent process harus menunggu beberapa detik (misalnya 5 detik) sebelum membaca status child process dan berakhir, menciptakan kondisi di mana child process menjadi Zombie sementara.
2. Child process harus segera berakhir setelah menciptakan sub-child process, menciptakan kondisi di mana sub-child process menjadi Orphan.
3. Sub-child process harus menampilkan pesan bahwa ia adalah Orphan dan menunggu beberapa detik (misalnya 10 detik) sebelum berakhir.
4. Program harus menampilkan PID dari masing-masing process untuk memudahkan identifikasi dan debug.

**Pertanyaan**

Tulislah kode dalam bahasa C yang memenuhi persyaratan di atas. Pastikan untuk menangani kondisi Zombie dan Orphan seperti yang dijelaskan. Gunakan fungsi-fungsi seperti **fork()**, **wait()**, **getpid()**, dan **sleep()** untuk mengimplementasikan solusi Anda.

**Catatan**

Program yang Anda kembangkan harus dapat menunjukkan pemahaman Anda tentang konsep Zombie dan Orphan process dalam sistem operasi. Penting untuk menggunakan fungsi **wait()** atau fungsi serupa untuk menangani proses Zombie dan memastikan semua proses berakhir dengan benar. Program harus rapi, dengan komentar yang memadai untuk menjelaskan alur kerja dan logika di balik kode Anda.

**UJIAN TENGAH SEMESTER GENAP TAHUN AKADEMIK 2023/2024**  
**PROGRAM STUDI INFORMATIKA**  
**FAKULTAS TEKNIK DAN INFORMATIKA**  
**UNIVERSITAS MULTIMEDIA NUSANTARA**

**Rubrik Penilaian**

Aspek/Kriteria yang Dinilai	Kriteria Penilaian				
	Sangat Kurang	Kurang	Cukup	Baik	Sangat Baik
	1	3	5	8	10
<i>Source code</i>	Tidak ada yang dikumpulkan/ kode terindikasi plagiarisme.	Ada kode yang dikumpulkan, tetapi <i>corrupt</i> , tidak dapat dikompilasi, atau tidak dapat dieksekusi.	Dapat dikompilasi dan dieksekusi, tetapi hasilnya salah semua.	Dapat dikompilasi dan dieksekusi, program menghasilkan keluaran yang tidak sepenuhnya salah.	Dapat dikompilasi, dieksekusi dan keluaran yang dihasilkan sesuai.

Acuan Pembuatan Soal:	Soal dibuat oleh:	Disetujui oleh:
<b>Week 1-7 IF260 Sistem Operasi</b>	<b>a.n Tim Dosen Mata Kuliah</b>  <b>(Suwito Pomalingo, S.Kom., M.Kom)</b> <b>Koord. Mata Kuliah</b>	<b>(Yaman Khaeruzzaman, M.Sc)</b> <b>PJS Ketua Program Studi</b>