Exercícios Experimento 3

**Pergunta 1:** Uma região por ser crítica tem garantida a exclusão mútua?

**Resposta:** Não se pode esquecer que Região Crítica é a área de um código ou recurso compartilhado que depende expressivamente que o acesso à mesma seja realizado de maneira sequencial. Porém, sabendo da possibilidade de condição de corrida, essa região deverá ser tratada de maneira a evitar/tratar condições de corrida.

Com isso, faz-se necessário a implementação da Exclusão Mútua que, por sua vez, consiste em métodos que podem ser criados (via implementação de seméforos, monitores, flags e afins) a fim de evitar o problema de condição de corrida e, de certa maneira, a possibilidade de outros processos invadirem uma região crítica.

**Pergunta 2:** É obrigatório que todos os processos que acessam o recurso crítico tenham uma região crítica igual?

**Resposta:** Depende, Se nenhum processo está executando dentro de sua seção crítica e existem alguns processos que desejam adentrar suas seções críticas, então a seleção do próximo processo que irá entrar em sua seção crítica não pode ser adiada. Ou seja, se os processos são de recursos compartilhados entre si eles possuem mesmo recurso crítico, já os não compartilhados, não dividem o mesmo recurso crítico, pois os dados estão em memórias diferentes.

**Pergunta 3:** Porque as operações sobre semáforos precisam ser atômicas?

**Resposta:** Essa obrigação evita condições de disputa entre vários processos, visando com que a exclusão mútua não seja violada.

**Pergunta 4:** O que é uma diretiva ao compilador?

**Resposta:** Em C, existem comandos que são processados durante a **compilação** do programa. Estes comandos são genericamente chamados de diretivas de compilação. Estes comandos informam ao compilador do C basicamente quais são as **constantes simbólicas** usadas no programa e quais **bibliotecas** devem ser anexadas ao programa executável.  Existem diversas diretivas como:

* **Compilação Conditional**- Usando diretivas de pré-processamento especiais, você pode incluir ou excluir partes do programa de acordo com várias condições;
* **Controle de Linha** - Se você usa um programa para combinar ou rearranjar os arquivos de origem em um arquivo intermediário, que será compilado, você pode usar o controle de linha para informar o compilador de onde veio cada linha;
* **Relatório de Erros e de advertência** - A diretiva **'#error'** faz com que o pré-processador Informe um erro fatal e a diretiva '**#warning**' faz com que a o pré-processador emita um aviso e continue pré-processamento.

**Pergunta 5:** Porque o número é pseudo aleatório e não totalmente aleatório?

**Resposta:** O número é Pseúdo-Aleatório pois visa-se um padrão na variação de alocação de mensagem e no tempo gasto para a troca de dados.

Questões Código:

**Pergunta 1:** Se usada a estrutura g\_sem\_op1 terá qual efeito em um conjunto de semáforos?

**Resposta:** A estrutura g\_sem\_op1 serve para realizar o controle das operações de semáforos. Ela é uma struct com os recursos capazes de guardar o identificador do semáforo (g\_sem\_op1[].sem\_num), o estado do semáforo (g\_sem\_op1[].sem\_flg) como fechado ou aberto, e por fim a operação do semáforo (g\_sem\_op1[].sem\_op), que é a necessidade dele liberar um processo ou fechar sua região crítica para que outro processo não acesse simultaneamente o que este está acessando.

**Pergunta 2:** Para que serve esta operação semop(), se não está na saída de uma região crítica?

**Resposta:** A função semop() permite efetuar operações sobre os semáforos identificados por semid, caso o semop() esteja fora de uma região crítica significa que o semáforo está aguardando um recurso de um novo processo.

**Pergunta 3:** Para que serve essa inicialização da memória compartilhada com zero?

**Resposta:** A inicialização da memória com zero acontece quando é possível gerar o segmento de memória compartilhada. Dessa forma, é endereçado um segmento de memória lógico com valor 0 para posteriormente ser utilizado no segmento de memória físico.

**Pergunta 4:** se os filhos ainda não terminaram, semctl e shmctl, com o parametro IPC-RMID, não permitem mais o acesso ao semáforo / memória compartilhada?

**Resposta:** Sim, caso ocorra a chamada das funções semctl e shmctl os semáforos e o acesso a memória compartilhada serão destruídos. Dessa forma, quando os filhos forem acessar os dados na mémoria não haverá essa estrutura de controle para evitar a ocorrência de race condition.

**Pergunta** 5: quais os valores possiveis de serem atribuidos a number?

**Resposta:**