



## PRIMEIRO TRABALHO PRÁTICO

### 1. Enunciado

Desenvolva um simulador de sistemas operacionais que tenha as seguintes funcionalidades: gerenciamento de processos, gerenciamento de memória e gerenciamento de dispositivos de entrada e saída. O simulador será composto por um microkernel que implementará as chamadas de sistema `send()` e `receive()` e se comunicará com os outros processos do SO. Quando um processo for criado será feita uma notificação via chamada de sistema ao kernel e este encaminhará o pedido para ao processo gerenciador de processos, este por sua vez solicitará ao gerenciador de memória uma área de memória livre que comporte o processo. Se o processos necessitar acessar algum dispositivo de E/S deverá solicitar formalmente ao gerenciador de dispositivos de E/S que permitirá ou não o acesso ao recurso.

### 2. Problema

O simulador deve ser desenvolvido em C com o uso das bibliotecas `unistd.h`, `pthread.h`, `semaphore.h`. O microkernel deverá se comunicar com o gerenciador de processos, com o gerenciador de memória e com o gerenciador de dispositivos de E/S usando `socket` ou `pipe` nomeado. Cada processo novo criado deverá ser uma thread dentro do gerenciador de processos que será representado por uma PCB contendo informações sobre o processo. Não se esqueça de sincronizar as ações das threads com o uso de semáforos, pois estas necessitarão de informações que são comuns a todas as demais threads.

### 3. Informações

O trabalho deverá ser feito em dupla ou individualmente. No dia da apresentação, data que deverá ser marcada com antecedência com o professor, o aluno ou alunos irão apresentar o programa em execução mostrando as funcionalidades do mesmo, bem como apresentar o código fonte. As notas serão atribuídas de maneira individual, ou seja, caso o trabalho tenha sido feito em dupla cada um dos autores receberá uma nota. Esta nota dependerá do desempenho durante apresentação do trabalho onde o professor arguirá o aluno. O trabalho deverá ser desenvolvido na linguagem de programação C usando as bibliotecas de semáforos, threads e comunicação entre processos (`socket` local ou `pipe` nomeado).

### 4. Datas Importantes

- **29/11/2014:** data final para postar o trabalho no sistema Moodle.
- **01/12/2014:** data final para a apresentação do trabalho. Marque um dia e horário com o professor para fazer a apresentação do trabalho.