

Trabalho 1 – Sistemas Operacionais 2 (INE5424) – INE – UFSC  
Matheus Bratfisch – 08132038  
Olav P. Henschel – 08132039

O trabalho 1, possui como proposta a alteracao do metodo “join” das Threads. Na versao inicial o sistema basicamente, possui um while, verificando o estado da thread que se esta dando join, caso seja diferente de FINISHING, simplesmente utiliza o metodo yield para parar sua execucao. Este metodo e considerado ineficiente ja que o mesmo fica em execucao o tempo todo. Saindo para a fila de \_ready sempre que o metodo yield fosse chamado. Fazendo assim inumeras verificacoes do estado do processo.

Inicialmente, foi imaginado como solucao para este problema a criacao de uma nova variavel no arquivo thread.h, esta denominada de \_waiting\_for\_me. Onde ao se chamar o metodo join para um Thread a thread que esta chamando o metodo join define esta variavel com um ponteiro para si mesmo. Em seguida a thread em execucao se auto suspende. Quando a Thread que foi dado join for chamado seu metodo exit, a mesma faz a verificacao da variavel, caso a mesma possua algum valor, ela acorda o processo que esta sendo apontado pela variavel.

Infelizmente pensamos que esta solucao nao seria viavel ja que, mais de um Thread poderiam dar Join, fazendo assim que criemos uma situacao de “deadlock”, onde a thread que deu join primeiro nunca seria acordada.

Foi constatado que deveria ser implementado uma fila com os processos, a utilizacao da variavel \_link foi utilizada, porem, assim foi criado outro problema ja que, a mesma estaria em duas filas. Fazendo com que as filas fiquem inconsistente. Porque o Elemento de fila so pode estar em uma fila por vez. A utilizacao de outro elemento de fila, foi descartada ja que seria necessario criar mais uma fila por Thread, causando um overhead desnecessario.

Pensando em como resolver este problema, fomos pesquisar como funciona o metodo join em outros sistemas. Durante esta pesquisa foi descoberto que pelo padrao POSIX, uma Thread que nao a pai, dar Join eh um erro de logica. Entao chegamos a conclusao que a primeira solucao comentada, realizaria a tarefa indicada de maneira satisfatoria.