**Grupo :**

Newton Coelho - 1621351

Hugo de Barros Pimentel Dutra Machado - 1410530

**Relatório:**

**Introdução :**

O nosso programa contem um arquivo interpretador , um arquivo escalonador e 5 arquivos de programas de usuários . O motivo pelo qual criamos 5 programas usuários foi dar a possibilidade do usuário poder definir o número de rajadas possíveis ,de 1 a 5,

de cada processo que irá entrar no escalonador.

**Programas do usuário:**

Para executar o programa , primeiro tem que se criar 5 programas usuários compilando da seguinte forma: programa-usuario1.c com o nome programa-usuario1 , programa-usuario2.c com o nome programa-usuario2, programa-usuario3.c com o nome programa-usuario3 , programa-usuario4.c com o nome programa-usuario4 e programa-usuario5.c com o nome programa-usuario5 , e compilar os arquivos interpretador.c e mqlf2.c(escalonador).

**Interpretador :**

O arquivo de interpretador lê o numero de rajadas e os valores das rajadas em si passadas no terminal da seguinte forma :

O usuário deve digitar entre (1 e 5) no primeiro argumento do terminal , e digitar o número de rajadas na forma XXX(X sendo um número) nos seguintes argumentos do terminal para dar início a um programa.

Exemplo :

1 001

2 020 010

3 040 050 040

**MLQF2(Escalonador)**

O arquivo mlqf2 é o escalonador de processos requisitado no trabalho. O usuário vai inserindo programas no interpretador e o escalonador vai mostrando o pid do processo daquele programa que se encontra com a maior prioridade com base no número de processos em cada fila( Comor requerido no enunciado do trabalho).

Execução do programa com o arquivo mlqf2.c :

Você deve executar os módulos interpretador.c e mlqf2.c. Durante a execução do modulo mlqf2 , o pid do processo com maior prioridade ira aparecer a cada segundo , o número de processos em cada fila , juntamente com o contador de tempo a cada segundo baseado no quantum da fila do processo atual , e as mensagens ao entrar em io(Quando termina uma rajada) , ao cair de uma fila , ao subir de uma fila e a mensagem de fim do processo.

O mlqf2.c executa o Round Robin de forma correta , botando no final da fila de processos cada novo processo que entra com as rajadas determinadas , demonstrando qual processo ganha a vez(está sendo executado), trocando de processos caso a rajada do processo atual seja maior que o quantum de tempo e possua processo de maior prioridade, e também entrando em modo IO quando acaba uma rajada . Caso um processo entre em io , o processo seguinte com maior prioridade vai executar no tempo de 3 segundos(Travada do io) , e depois caso o processo anterior ainda possua rajadas e ainda seja o com maior prioridade , ele volta a executar logo após os 3 segundos , travando o processo que tinha acabado de entrar.

**Conclusão :**

Concluímos que o trabalho foi feito de forma completa , sendo possível criar os programas de usuário e ver o funcionamento do escalonador e do interpretador de forma completa.