Sistemas Operativos I

Escalonador Round Robin

Parte II

Docentes: Professores Luís Rato e Nuno Miranda

UNIVERSIDADE DE ÉVORA

Trabalho realizado por:

-Alexandre Rodrigues 35449

-Pedro Oliveira 35480

Introdução

No âmbito da disciplina de Sistemas Operativos I, foi-nos pedido que realizássemos, na linguagem C, um escalonador Round Robin de cinco estados.

Descrição das funções

Queue.c

-struct node *newNode(struct pcb *p, struct list *lista)

Recebe um PCB, criando um nó com ele.

-struct list *newLista()

Cria uma nova queue.

-void insert(struct pcb *p, struct list *newLista)

Recebe um PCB e uma lista, e insere esse PCB no princípio dessa lista.

-void printLista(struct list *newLista)

Faz o print da lista.

-struct pcb *top(struct list *newLista)

Recebe uma lista e devolve o Header dessa mesma lista, o qual é um PCB.

-struct pcb *remover(int v, struct list *newLista)

Recebe um inteiro (que será o top), e remove o PCB com esse ID da lista também recebida.

-bool empty(struct list *newLista)

Recebe uma lista e verifca se está vazia ou não.

-struct pcb *search(int v,struct list *newLista)

Recebe um inteiro (o ID que pretendemos encontrar), uma lista, e devolve o PCB com o ID que queremos retirar.

Scheduler.c

-void newToReady()

Remove o top da queue NEW, e insere na READY.

-void blockToReady()

Remove o top da queue BLOCK, e insere na READY.

-void readyToRun()

Remove o top da queue READY, e insere no RUN.

-void runToReady()

Remove o top da queue RUN, e insere na READY.

-void runToBlock()

Remove o top da queue RUN, e insere no BLOCK.

-struct pcb *newPcb(int prcId)

Recebe um inteiro, que é o ID do processo, e devolve um PCB.

A struct pcb *newPcb contém estes componentes:

-int ini

Início do processo.

-int fim

Fim do processo.

-int pc

Instrução onde se encontra.

-int id

Identificador único.

-int nBegin

Número de vezes que já fez um GOTOBEGIN10.

-int qRun

Verifica o Quantum no RUN.

-int qBlock

Verifica o número de ciclos no BLOCK.

-int estado

Guarda o estado em que está atualmente.

-void ciclo()

Função onde se realizam todas as verificações de prioridade e ações do scheduler.

-void writeInFile(FILE *nfp, struct list *ordem, int tempo)

Recebe um ficheiro chamado nfp, uma lista, um inteiro, e escreve no ficheiro.

-int main(void)

Inicializa todas as queues e arrays que precisamos, de seguida pergunta se queremos o processo em modo WORST FIT ou BEST FIT. No fim da seleção introduzimos o nome do ficheiro, e podemos inserir mais caso queiramos, caso contrário inserimos exit. No caso de o ficheiro não existir, o programa devolve uma mensagem que diz que o ficheiro não existe, pedindo novamente um ficheiro. Depois é verificado se existe memória suficiente no array para introduzir um ficheiro, caso não haja, devolve a seguinte mensagem “Memória cheia – Impossível criar processo.”. Se, ao introduzir este ficheiro, o número máximo de ficheiros no programa tiver sido atingido, devolve a seguinte mensagem “Número máximo de ficheiros atingido.”

Observações

Dentro da função ciclo, o pedaço de código que faz o Fork não funciona para a segunda parte do trabalho. Como nos também melhorámos a primeira parte, está comentado o Fork da mesma.