Relatório do Simulador de Gestão de Processos

ID do grupo: 21
Elementos do grupo:
- **41059 - Rúben Guilherme**
- **41853 - Paulo Duarte**
- **41872 - Rui Barata**

GIT: https://gitlab.com/the-releasers/so-project

Índice

Introdução

Neste projeto pretende-se desenvolver um simulador de gestão e execução de processos que tem por base os conhecimentos adquiridos na unidade curricular de Sistemas Operativos na Universidade da Beira Interior.

Com este projeto para além de ser pretendido o reforço dos principais temas da unidade curricular em questão (como por exemplo: os algoritmos de escalonamento de processos, de gestão de memória e a forma como o processador os gere), também é importante a evolução das nossas capacidades de programação num projeto desta dimensão.

Resumo do estado do projeto

Atualmente o simulador já possui as funcionalidades base, permitindo simular o escalonamento não preemptivo com o algoritmo First Come First Serve. Desta forma consideramos ter concluído os objetivos pretendidos para a primeira entrega.

As funcionalidades operacionais são: um menu com opções que permitem alterar fluir da simulação (p. ex: o time quantum), um modo de debug que permite comandar o simulador através do terminal, com os comandos pré-estabelecidos, entre outras.

Posteriormente, iremos implementar mais algumas funcionalidades, nomeadamente: outros algoritmos de escalonamento (p. ex. o Shortest Job First tanto não preemptivo como o preemptivo, entre outros), alguns algoritmos de gestão de memória, um report mais completo e outras funcionalidades extra.

Módulos

Exec

Funções

Nome da função	execute
Inputs	i: int (índice do processo em execução)
Descrição	Executa uma instrução do processo em execução.
Outputs	unit ()
Efeito Secundário	No caso de este processo ter sido bloqueado ou terminado, então a função irá adicioná-lo à Queue dos processos bloqueados ou à Queue dos processos terminados, conforme o seu estado.

Report

Funções

Nome da função	print_queue
Inputs	process_queue : pcb Queue
Descrição	Esta função itera sobre a Queue que foi passada como argumento fazendo print dos atributos escolhidos de cada processo.
Outputs	unit ()
Efeito Secundário	Nenhum efeito secundário.

Nome da função	report
Inputs	unit ()
Descrição	Faz o report da simulação mostrando todas as informações pertinentes no momento em que é chamada.
Outputs	unit ()
Efeito Secundário	Nenhum efeito secundário.

Nome da função	turnaround
Inputs	unit ()
Descrição	Calcula o turnaround médio de todos os processos terminados.
Outputs	float
Efeito Secundário	Nenhum efeito secundário.

Nome da função	tme
Inputs	unit ()
Descrição	Calcula o tempo médio de espera de todos os processos terminados.
Outputs	float
Efeito Secundário	Nenhum efeito secundário.

Nome da função	burst_time
Inputs	unit ()
Descrição	Calcula o burst time médio de todos os processos.
Outputs	float
Efeito Secundário	Nenhum efeito secundário.

Nome da função	global_report
Inputs	unit ()
Descrição	Dá print das informações globais à simulação mais pertinentes (WIP).
Outputs	unit ()
Efeito Secundário	Nenhum efeito secundário.

Sim

Funções

Nome da função	line_to_instr
Inputs	line : string (instrução de um processo)
Descrição	Converte uma instrução : string numa instrução : instruct.
Outputs	instr
Efeito Secundário	Nenhum.

Nome da função	copy_process
Inputs	p : pcb
Descrição	Faz uma cópia do processo passado pelo argumento.
Outputs	pcb
Efeito Secundário	Nenhum efeito secundário.

Nome da função	find_process_name
Inputs	s : string ; i : int
Descrição	A função deve ser chamada com i = 0. Percorre a pcb_table e se encontrar um processo com o nome igual a “s” então devolve o indice do processo. Se não encontrar devolve -1.
Outputs	int
Efeito Secundário	Nenhum efeito secundário.

Nome da função	openfile
Inputs	newP : newP
Descrição	Se o processo já existir na tabela de processos, então atualiza os seus atributos e põe o processo na tabela de ready. Se não abre o ficheiro de texto .prg onde se encontram as instruções a executar pelo processo, põe as instruções em memória e insere este processo numa tabela de processos e numa tabela de processos ready.
Outputs	unit ()
Efeito Secundário	A seguir à abertura do ficheiro .prg existe a criação do processo : pcb para que seja possível inserí-lo na tabela de processos.

Nome da função	openfile_string
Inputs	str : string
Descrição	Abre o ficheiro .prg e põe as instruções do processo em questão em memória.
Outputs	unit ()
Efeito Secundário	Nenhum efeito secundário.

Nome da função	read_instr
Inputs	process : pcb
Descrição	Procura a instrução que o processo pretende efetuar e executando-a.
Outputs	unit ()
Efeito Secundário	Dependendo da instrução escolhida, os efeitos secundários irão variar.

Long

Funções

Nome da função	unblock
Inputs	v : boolean array (array em que cada indice representa um processo bloqueado) n : número de processos bloqueados
Descrição	Percorre o array v e se encontrar um elemento a true então retira o processo bloqueado com esse indice da blocked queue e passa-o para a ready queue. Existe sempre pelo menos um elemento a true
Outputs	unit ()
Efeito Secundário	Altera a blocked queue, retirando no mínimo um processo e passando-o para a ready queue

Nome da função	long_sched
Inputs	n : número de processos bloqueados
Descrição	Esta função escolhe de forma aleatória um processo para desbloquear (cada um tem 1/n chances de ser desbloqueado), havendo sempre pelo menos um processo a desbloquear. De seguida chama a função unblock.
Outputs	unit ()
Efeito Secundário	Chama a função unblock

Main

Funções

Nome da função	clock
Inputs	unit ()
Descrição	Um while com uma flag que chama todas as funções que fazem com que o simulador funcione. O clock é chamado no menu principal e apenas termina quando o gestor de processos encontra a instrução T no control.txt ou no stdin (dependendo do modo da simulação).
Outputs	unit ()
Efeito Secundário	Diversos efeitos secundários, visto ser a função principal

Nome da função	menu
Inputs	unit ()
Descrição	Menu com 3 opções: começar a simulação, abrir o menu de opções ou sair do programa. Este menu é a primeira função chamada no princípio da execução do simulador.
Outputs	unit ()
Efeito Secundário	Começar o clock; Sair do programa.

Nome da função	options_menu
Inputs	unit ()
Descrição	Menu com 4 opções: entrar no menu de escolha de algoritmo de escalonamento, alterar o tipo de escalonamento, alterar o time quantum do programa e voltar ao menu principal
Outputs	unit ()
Efeito Secundário	Nenhum efeito secundário.

Nome da função	scheduling_menu
Inputs	unit ()
Descrição	Menu com várias opções que permitem escolher o algoritmo de escalonamento a usar. De momento apenas é possível escolher o First Come First Serve.
Outputs	unit ()
Efeito Secundário	Nenhum efeito secundário.

Process

Funções

Nome da função	read_plan
Inputs	unit ()
Descrição	Abre o ficheiro plan.txt e introduz a informação daí retirada na newP - fila de processos novos.
Outputs	unit ()
Efeito Secundário	Nenhum efeito secundário.

Nome da função	interrupt
Inputs	unit ()
Descrição	Bloqueia o processo em execução
Outputs	unit ()
Efeito Secundário	Nenhum efeito secundário.

Nome da função	read_command
Inputs	c - char (char que representa o comando a ler)
Descrição	Interpreta o comando recebido (c). ‘E’ - Executa um processo durante o time_quantum; ‘I’ - Interrompe o processo em execução; ‘D’ - Faz o escalonamento a longo prazo, desbloqueando processos; ‘R’ - Faz um report do estado atual do simulador ‘T’ - Termina o simulador, colocando a clock_flag a false e faz o report das estatísticas globais
Outputs	unit ()
Efeito Secundário	Depende do comando recebido

Nome da função	read_terminal
Inputs	unit ()
Descrição	Lê uma instrução do stdin e chama o read_command
Outputs	unit ()
Efeito Secundário	Recebe uma linha do stdin

Nome da função	read_control
Inputs	fi - in_channel (Canal de input para o ficheiro)
Descrição	Lê uma instrução do fi (control.txt) e chama o read_command
Outputs	unit ()
Efeito Secundário	Nenhum efeito secundário.

Nome da função	controller
Inputs	unit ()
Descrição	Se o simulador está em modo debug chama o read_terminal, se não está em modo debug chama o read_control
Outputs	unit ()
Efeito Secundário	Nenhum efeito secundário.

Variáveis

Nome da Variável	fc
Descrição	Canal de input usado para ler do ficheiro “control.txt”
Tipo	in_channel

Nome da Variável	buffercommand
Descrição	Caracter usado quando é necessário guardar em buffer uma instrução. Quando não há nada em buffer tem armazenado o dollar sign ($).
Tipo	char

LIB

Funções

Nome da função	remove_CR
Inputs	str : string
Descrição	Recebe uma string e remove o último caracter se esse caracter for ‘\r’. Devolve a string sem esse caracter.
Outputs	string
Efeito Secundário	Nenhum efeito secundário.

Types

Nome do Type	running_state	
Descrição	Type que vai conter os dados dos processos em execução.	
Variáveis	ind : int	Índice da tabela de pcbs (pcb_tabela) do processo que está a correr.
	pid : int	Pid do processo que está a correr.
	pc : int	Pc do processo que está a correr.

Nome do Type	instruction	
Descrição	Type que vai ser usado para construir a memória.	
Variáveis	ins: char	Caracter usado para representar o tipo de instrução.
	n : int	Número usado para o caso da instrução ter um inteiro.
	name : string	String usada o caso da instrução ter uma string.

Nome do Type	newP	
Descrição	Type que contém as informações lidas de uma linha do plano.txt.	
Variáveis	name : string	String com o nome do programa
	time : int	Int que contém o tempo a que o processo deve ser criado e inserido na ready queue
	priority : int	Int que contém a prioridade do processo

Nome do Type	pcb	
Descrição	Type que contém os dados necessários para construir um processo.	
Variáveis	name : string	String com o nome do processo
	start : int	Int com o primeiro endereço que o processo está em memória.
	variable : int	Int que contém o valor da variável do processo.
	pid : int	Int que contém o pid do processo
	ppid : int	Int que contém o ppid do processo
	priority : int	Int que contém a prioridade do processo
	arrival_time : int	Int que contém o tempo de chegada do processo
	time : int	Int que contém o tempo que o processo esteve no CPU
	pc : int	Int que contém o program counter do processo
	status : int	Int que indica o status do processo | 0 -> ready | 1 -> running | 2 -> blocked | 3 -> terminated
	finish : int	Int que contém o tempo a que o processo terminou

Nome do Type	pcb_list
Descrição	Type que serve de construtor para a pcb_list

Variáveis

Nome da Variável	memory
Descrição	Array que serve de memória para o nosso simulador.
Tipo	instruction array

Nome da Variável	next_memory_index
Descrição	Inteiro usado para guardar a próxima posição livre da memória.
Tipo	int ref

Nome da Variável	clock_flag
Descrição	Boolean usado para terminarmos o programa .
Tipo	bool ref

Nome da Variável	executing_flag
Descrição	Boolean usado para determinarmos quando devemos usar a função execute.
Tipo	bool ref

Nome da Variável	rem_time
Descrição	Inteiro usado para determinarmos quanto tempo um programa deve ser executado
Tipo	int ref

Nome da Variável	time_flag
Descrição	Boolean usado para auxiliar na contagem do tempo.
Tipo	bool ref

Nome da Variável	debug_mode
Descrição	Boolean usado para determinar se o utilizador pediu para entrar no modo de debug.
Tipo	bool ref

Nome da Variável	time_flag
Descrição	Boolean usado para auxiliar na contagem do tempo.
Tipo	bool ref

Nome da Variável	son_flag
Descrição	Boolean usado para determinar se o processo atual é um filho de outro processo.
Tipo	bool ref

Nome da Variável	time
Descrição	Inteiro usado para contar o tempo.
Tipo	int ref

Nome da Variável	time_quantum
Descrição	Inteiro usado para determinar o time_quantum do simulador (tempo máximo que um processo deve ser executado).
Tipo	int ref

Nome da Variável	next_pid
Descrição	Inteiro usado para atribuir aos processos pids diferentes.
Tipo	int ref

Nome da Variável	pcb_table
Descrição	Lista que contém todos os processos que são criados quando o simulador está a correr.
Tipo	pcb list ref

Nome da Variável	newQ
Descrição	Queue que contém todas as informações que são lidas do plano.txt para, no tempo devido, serem criados processos.
Tipo	newP Queue.t

Nome da Variável	readyQ
Descrição	Queue que contém todos os processos que estão prontos a executar.
Tipo	pcb Queue.t

Nome da Variável	blockedQ
Descrição	Queue que contém todos os processos que foram bloqueados durante a execução do simulador.
Tipo	pcb Queue.t

Nome da Variável	terminatedQ
Descrição	Queue que contém todos os processos que terminaram durante a execução do simulador.
Tipo	pcb Queue.t

Nome da Variável	runnning_proc
Descrição	Variável que contém as informações do processo em execução.
Tipo	running_state

Short

Funções

Nome da função	findProcInd
Inputs	queue : pcb list, pid : int, count : int.
Descrição	Devolve o índice da pcb_tabela do processo que tem o pid igual ao pid recebido.
Outputs	int
Efeito Secundário	Nenhum efeito secundário.

Nome da função	ffcfs
Inputs	() : unit
Descrição	Se não existir nenhum processo a ser executado (running_proc.ind = -1) então tira um processo da readyQ, atualiza o estado do mesmo e atualiza as informações do running_proc.
Outputs	() : unit
Efeito Secundário	Tira um processo da readyQ.

Nome da função	short_sched
Inputs	() : unit
Descrição	Função que executa o escalonamento de curto prazo escolhido pelo utilizador.
Outputs	() : unit
Efeito Secundário	Depende da variável selected_scheduller.

Variáveis

Nome da Variável	selected_scheduller
Descrição	Inteiro usado para guardar a escolha do escalonamento de curto prazo do utilizador
Tipo	int ref

Ficheiros

Makefile

Ficheiro usado pelo comando ‘make’ para compilar todo o código do trabalho num só ficheiro executável. Utiliza também o OCamlMakefile, um projeto opensource no github que adapta o Makefile para poder compilar OCaml: https://github.com/mmottl/ocaml-makefile

control.txt

Ficheiro onde se encontram os comandos para a execução do simulador.

plan.txt

Ficheiro onde se encontra a descrição dos processos a serem criados no simulador. O formato é: “name time_to_create priority” em cada linha.

nome_do_programa.prg

Programas a serem executados pelo simulador. O formato é “ins n/name” (ins : char; n : int; name : string;).

Conclusão:

WIP