



UNIVERSIDADE FEDERAL DO MARANHÃO
BACHARELADO INTERDISCIPLINAR EM CIÊNCIA E TECNOLOGIA
SISTEMAS OPERACIONAIS
TURMA 1

ELLEN CRISTINA DE SOUSA CASTRO
BRUNO KAUAN RODRIGUES SILVA
JOÃO VICTOR LIMA SALOMÃO
LETÍCIA BORGES ASSUNÇÃO
SAMARA SANTOS VIEGAS

MANUAL DO USUÁRIO
ALGORITMO DE ESCALONAMENTO DE PROCESSOS FIFO, SJR E SRTN

SÃO LUIS – MA

2024

Introdução

Este manual detalha o uso e a instalação de um aplicativo desenvolvido em Python, utilizando as bibliotecas tkinter para a interface gráfica e matplotlib para a visualização dos gráficos de Gantt. O aplicativo permite a gestão e escalonamento de processos utilizando os algoritmos FIFO, SJF e SRTN.

Explicação das Funções do Gerenciador de Processos

Classe Processo

A classe Processo define a estrutura de um processo, incluindo seu identificador, tempo de duração, tempo de chegada, e outros atributos relacionados à sua execução.

- **__init__**: Inicializa um objeto Processo com atributos como pid (identificador do processo), duracao (tempo de duração), chegada (tempo de chegada), entre outros.

Função adicionar_processo

Adiciona um novo processo à lista de processos.

- Obtém os valores das entradas de texto (PID, duração, chegada), cria um novo objeto Processo, adiciona-o à lista e atualiza a exibição da lista de processos. Em caso de erro nos valores inseridos, exibe uma mensagem de erro.

Função remover_processo

Remove o processo selecionado da lista de processos.

- Obtém o índice do processo selecionado na lista e remove-o. Atualiza a lista exibida e exibe uma mensagem de sucesso ou erro.

Função executar_fifo

Executa o algoritmo de escalonamento FIFO (First In, First Out).

- Ordena os processos pela ordem de chegada e calcula os tempos de início, fim, espera, turnaround e resposta para cada processo. Atualiza a tabela de processos e plota o gráfico de Gantt correspondente.

Função executar_sjf

Executa o algoritmo de escalonamento SJF (Shortest Job First).

- Ordena os processos pelo tempo de duração e calcula os tempos de início, fim, espera, turnaround e resposta para cada processo. Atualiza a tabela de processos e plota o gráfico de Gantt correspondente.

Função executar_srtm

Executa o algoritmo de escalonamento SRTN (Shortest Remaining Time Next).

- Ordena os processos por ordem de chegada e, a cada unidade de tempo, escolhe o processo com menor tempo restante para executar. Calcula e atualiza os tempos de início, fim, espera, turnaround e resposta conforme necessários. Atualiza a tabela de processos e plota o gráfico de Gantt correspondente.

Função atualizar_lista_processos

Atualiza a lista de processos exibida na interface.

- Remove todos os itens da lista e insere os processos atuais com suas informações.

Função limpar_entradas

Limpa as entradas de texto da interface.

- Remove qualquer texto presente nos campos de entrada de PID, duração e chegada.

Função atualizar_tabela

Atualiza a tabela de processos exibida na interface.

- Remove todas as linhas da tabela e insere os processos ordenados com seus respectivos valores.

Função plot_grafico_fifo

Plota o gráfico de Gantt para o algoritmo FIFO.

- Cria barras representando os tempos de início e fim de cada processo, exibindo o gráfico com cores diferentes para cada processo e rótulos correspondentes.

Função plot_grafico_sjf

Plota o gráfico de Gantt para o algoritmo SJF.

- Similar ao gráfico de Gantt para FIFO, mas para processos ordenados pelo tempo de duração.

Função plot_grafico_srtm

Plota o gráfico de Gantt para o algoritmo SRTN.

- Cria barras representando cada unidade de tempo para cada processo na sequência de execução, exibindo o gráfico com cores diferentes para cada processo e rótulos correspondentes.

Função executar_escalador

Executa o algoritmo de escalonamento selecionado.

- **Descrição:** Obtém o algoritmo selecionado na combobox (FIFO, SJF, SRTN) e chama a função correspondente para executar o algoritmo de escalonamento.

Interface Gráfica

A interface gráfica foi criada usando a biblioteca tkinter do Python.

- **Janela Principal:** A janela principal (janela) é configurada com um título e um estilo visual para os elementos da interface.
- **Frame Principal:** Um frame (frame_principal) é utilizado para organizar os widgets (entradas de texto, botões, labels, etc.) de forma estruturada.
- **Entradas de Texto:** Campos de entrada de texto para PID, duração e chegada dos processos.
- **Botões:** Botões para adicionar, remover processos e executar o algoritmo de escalonamento selecionado.
- **Lista de Processos:** Um Listbox para exibir os processos adicionados.
- **Combobox:** Uma combobox para selecionar o algoritmo de escalonamento a ser executado.
- **Tabela:** Uma tabela (Treeview) para exibir os dados dos processos após a execução do algoritmo.
- **Mensagens de Resultado:** Labels para exibir mensagens de resultado ou erro.

Requisitos

- **Python:** Versão 3.6 ou superior.
- **Bibliotecas Python:**
 - tkinter
 - matplotlib

Instalação das Bibliotecas

Para instalar as bibliotecas necessárias, execute os seguintes comandos:

```
pip install matplotlib
```

```
pip install tkinter
```

Instalação do Algoritmo

1. **Clone ou Baixe o Código:**
 - Clone o repositório ou baixe o arquivo Python diretamente.
2. **Execute o Script:**
 - Abra um terminal ou prompt de comando.
 - Navegue até o diretório onde o arquivo gerenciador_processos.py está salvo.
 - Execute o script usando Python:

```
python gerenciador_processos.py
```

Uso do Algoritmo

Interface Gráfica

A interface gráfica é composta por diversos elementos, incluindo entradas de dados, botões, listas e tabelas como visto na figura a seguir:

The screenshot shows a window titled "Gerenciador de Processos" with a standard Windows-style title bar (minimize, maximize, close buttons). The interface is divided into several sections:

- Input Fields:** Three text boxes on the left are labeled "PID do Processo:", "Tempo de Duração do Processo:", and "Tempo de Chegada do Processo:". To their right are three corresponding empty text input fields.
- Action Buttons:** Two blue buttons with white text are centered: "Adicionar Processo" and "Remover Processo".
- Algorithm Selection:** Below the buttons is a large, empty rectangular box. Further down, the label "Algoritmo de Escalonamento:" is followed by a dropdown menu currently showing "FIFO".
- Execution Button:** A blue button with white text labeled "Executar Escalonamento" is positioned below the dropdown.
- Table:** At the bottom, there is a table with six columns: "PID", "Duração", "Chegada", "Espera", "Turnaround", and "Resposta". The table body is currently empty.

Veja a seguir a descrição detalhada de cada componente:

1. **Entrada de Dados:**

- **PID do Processo:** Campo para inserir o identificador único do processo.
- **Tempo de Duração:** Campo para inserir a duração do processo em unidades de tempo.
- **Tempo de Chegada:** Campo para inserir o tempo de chegada do processo.

2. **Botões:**

- **Adicionar Processo:** Adiciona um novo processo à lista.
- **Remover Processo:** Remove o processo selecionado da lista.
- **Executar Escalonamento:** Executa o algoritmo de escalonamento escolhido.

3. **Lista de Processos:**

- Exibe os processos adicionados, mostrando seu PID, duração e tempo de chegada.

4. **Combobox para Seleção do Algoritmo:**

- Permite escolher entre os algoritmos FIFO, SJF e SRTN.

5. **Tabela de Processos:**

- Exibe os detalhes dos processos, incluindo PID, duração, chegada, tempo de espera, turnaround e tempo de resposta.

6. **Área de Resultados:**

- Mostra mensagens de resultado ou erros após a execução dos algoritmos.

Funcionalidades

- **Adicionar Processo:**
 - Insira os valores nos campos correspondentes e clique em "Adicionar Processo".
- **Remover Processo:**
 - Selecione um processo na lista e clique em "Remover Processo".
- **Escolher Algoritmo e Executar:**

- Selecione o algoritmo desejado no combobox e clique em "Executar Escalonamento".

Algoritmos de Escalonamento

1. FIFO (First-In-First-Out):

- Executa os processos na ordem de chegada.

2. SJF (Shortest Job First):

- Executa os processos com menor duração primeiro.

3. SRTN (Shortest Remaining Time Next):

- Executa o processo com menor tempo restante, ajustando a execução dinamicamente.

Exemplos de Uso

Adicionar um Processo

- **Dados:**
 - PID: 1
 - Duração: 5
 - Chegada: 0
- **Ação:**
 - Insira os dados nos campos e clique em "Adicionar Processo".

Remover um Processo

- **Ação:**
 - Selecione um processo na lista e clique em "Remover Processo".

Executar FIFO

- **Ação:**
 - Selecione "FIFO" no combobox e clique em "Executar Escalonamento".

Executar SJF

- **Ação:**
 - Selecione "SJF" no combobox e clique em "Executar Escalonamento".

Executar SRTN

- **Ação:**
 - Selecione "SRTN" no combobox e clique em "Executar Escalonamento".

Visualização dos Gráficos

Após a execução dos algoritmos, os gráficos de **Gantt** são exibidos, mostrando a sequência e o tempo de execução de cada processo. Os gráficos são gerados com **matplotlib** e são apresentados em janelas separadas.

Erros Comuns e Soluções

- **Erro ao Adicionar Processo:** Verifique se todos os campos estão preenchidos corretamente com valores válidos.
- **Erro ao Remover Processo:** Certifique-se de que um processo está selecionado na lista.
- **Problemas com Gráficos:** Verifique se a biblioteca **matplotlib** está instalada corretamente.

RECONHECIMENTOS E DIREITOS AUTORAIS

@AUTOR: BRUNO KAUAN RODRIGUES SILVA, ELLEN CRISTINA DE SOUSA CASTRO, JOÃO VICTOR LIMA SALOMÃO, LETICIA BORGES ASSUNÇÃO E SAMARA SANTOS VIEGAS

@CONTATO: BRUNO.KAUAN@DISCENTE.UFMA.BR ELLEN.CASTRO@DISCENTE.UFMA.BR JOAO.SALOMAO@DISCENTE.UFMA.BR LETICIA.ASSUNCAO@DISCENTE.UFMA.BR SAMARA.VIEGAS@DISCENTE.UFMA.BR

@DATA ÚLTIMA VERSÃO: 18/06/2024

@VERSÃO: 1.0

@OUTROS REPOSITÓRIOS: [HTTPS://GITHUB.COM/BKAUAN099](https://github.com/BKAUAN099)

@AGRADECIMENTOS: UNIVERSIDADE FEDERAL DO MARANHÃO (UFMA), PROFESSOR DOUTOR THALES LEVI AZEVEDO VALENTE, E COLEGAS DE CURSO.

@COPYRIGHT/LICENSE

ESTE MATERIAL É RESULTADO DE UM TRABALHO ACADÊMICO PARA A DISCIPLINA SISTEMAS OPERACIONAIS, SOB A ORIENTAÇÃO DO PROFESSOR DR. THALES LEVI AZEVEDO VALENTE, SEMESTRE LETIVO 2024.1, CURSO ENGENHARIA DA COMPUTAÇÃO, NA UNIVERSIDADE FEDERAL DO MARANHÃO (UFMA). TODO O MATERIAL SOB ESTA LICENÇA É SOFTWARE LIVRE: PODE SER USADO PARA FINS ACADÊMICOS E COMERCIAIS SEM NENHUM CUSTO. NÃO HÁ PAPELADA, NEM ROYALTIES, NEM RESTRIÇÕES DE "COPYLEFT" DO TIPO GNU. ELE É LICENCIADO SOB OS TERMOS DA LICENÇA MIT REPRODUZIDA ABAIXO E, PORTANTO, É COMPATÍVEL COM GPL E TAMBÉM SE QUALIFICA COMO SOFTWARE DE CÓDIGO ABERTO. É DE DOMÍNIO PÚBLICO. OS DETALHES LEGAIS ESTÃO ABAIXO. O ESPÍRITO DESTA LICENÇA É QUE VOCÊ É LIVRE PARA USAR ESTE MATERIAL PARA QUALQUER FINALIDADE, SEM NENHUM CUSTO. O ÚNICO REQUISITO É QUE, SE VOCÊ USÁ-LOS, NOS DÊ CRÉDITO.

COPYRIGHT © 2024 EDUCATIONAL MATERIAL

ESTE MATERIAL ESTÁ LICENCIADO SOB A LICENÇA MIT. É PERMITIDO O USO, CÓPIA, MODIFICAÇÃO E DISTRIBUIÇÃO DESTE MATERIAL PARA QUALQUER FIM, DESDE QUE ACOMPANHADO DESTE AVISO DE DIREITOS AUTORAIS.

O SOFTWARE É FORNECIDO "COMO ESTÁ", SEM GARANTIA DE QUALQUER TIPO, EXPRESSA OU IMPLÍCITA, INCLUINDO MAS NÃO SE LIMITANDO ÀS GARANTIAS DE COMERCIALIZAÇÃO, ADEQUAÇÃO A UM DETERMINADO FIM E NÃO INFRINGÊNCIA. EM NENHUM CASO OS AUTORES OU DETENTORES DE DIREITOS AUTORAIS SERÃO RESPONSÁVEIS POR QUALQUER RECLAMAÇÃO, DANOS OU OUTRA RESPONSABILIDADE, SEJA EM AÇÃO DE CONTRATO, TORT OU OUTRA FORMA, DECORRENTE DE, FORA DE OU EM CONEXÃO COM O SOFTWARE OU O USO OU OUTRAS NEGOCIAÇÕES NO SOFTWARE.

Para mais informações sobre a Licença MIT: <https://opensource.org/licenses/MIT>