PONTIFÍCIA UNIVERSIDADE CATÓLICA DO RIO GRANDE DO SUL

FACULDADE DE INFORMÁTICA

Curso de BACHARELADO EM ENGENHARIA DE SOFTWARE

AGES - agência experimental de engenharia de software

FELIPE GUEDES

MATEUS HAAS

SEGUNDO TRABALHO DE SISTEMAS OPERACIONAIS

período 2018/1

Porto Alegre

2018

FELIPE GUEDES

MATEUS HAAS

SEGUNDO TRABALHO DE SISTEMAS OPERACIONAIS

período 2018/1

Trabalho apresentado à Faculdade de Informática da Pontifícia Universidade Católica do Rio Grande do Sul, como requisito parcial para aprovação na disciplina de Sistemas Operacionais.

Professor: AVELINO FRANCISCO ZORZO

Porto Alegre

2018

SUMÁRIO

1 INTRODUÇÃO 14

2 PROJETO “NOME DO PROJETO XXXX” 15

3 ORIENTAÇÕES DE ESCRITA 17

4 CONCLUSÃO 21

REFERÊNCIAS 22

GLOSSÁRIO 23

ANEXOS 25

# INTRODUÇÃO

O presente trabalho foi desenvolvido pelos alunos Felipe Guedes e Mateus Haas durante o primeiro semestre de 2018 para a disciplina de Sistemas Operacionais ministrada pelo professor Avelino Zorzo.

O trabalho tem a intenção te aprimorar o conhecimenrto prático sobre gerência de memória virtual. Para isso foi requisitado que os alunos um programa que simulasse o funcionamento de in gerente de memória. Foi disponibilizado para os alunos um arquivo que especificou os requisitos que deveriam ser cumpridos durante o desenvolvimento do mesmo. Segundo Zorzo ( 2018, p1 ), o programa deve.

1. O programa deve funcionar em dois modos: sequencial e aleatório

2. No modo sequencial ele lê a criação de processos, término de processos, alocação de memória e acessos seguindo uma lista de comandos conforme exemplo abaixo.

3. No modo aleatório o programa deve:

a. O programa deve criar um conjunto de threads para simular processos executando.

b. Cada processo possui um tamanho, que representa quantos bytes ele ocupa na memória.

c. O processo passa o tempo todo: solicitando acessos endereços aleatórios de memória.

d. O processo pode também alocar mais memória com uma probabilidade de I%. e. O processo termina com uma probabilidade de J%.

4. O gerente de memória deve alocar o número de páginas para o processo, relativo ao tamanho do processo.

5. Para cada acesso, é necessário verificar se a página do processo onde aquele endereço se encontra, está ou não presente na memória. Se estiver o “acesso” é realizado sem problemas. Se não estiver, então o gerente de memória deve ser acionado e um “dump” da memória deve ser realizado, as tabelas de páginas dos processos, a situação da memória (que processo está ocupando cada página), e o endereço que gerou o page fault.

6. Deve haver alguma forma de acompanhar (visualizar) o que está acontecendo no programa a cada solicitação ou liberação

Escolha da liguagem que o algoritmo seria implementado foi de livre escolha dos alunos, nesta oportunidade o algoritmo foi implementado em java devido sua alta legibilidade, facilidade de desenvolvimento e portabilidade em comparação com outras linguagens como C.

# Arquivo de entrada

Nas especificações fornecedas aos alunos foi detalhado o formato do arquivo de entrada que os alunos receberiam para testar o funcionamento do programa.

Na primeira linha do arquivo estara

# Implementação dos algoritmos

4 CONCLUSÃO

REFERÊNCIAS

**Glossário (opcional)**

GLOSSÁRIO

Palavra – Significado da palavra

**Apêndice(s) (**Este item é elaborado pelo próprio autor do trabalho e serve para complementar a sua argumentação. É um elemento **opcional).**

APÊNDICES

APÊNDICE A – Análise dos relatórios mensais de uso do serviço de renovação de empréstimos.

APÊNDICE B – Análise dos relatórios mensais de uso do serviço de empréstimo domiciliar.

**Anexos (Este item é constituído por documentos complementares ao texto do trabalho e que não são elaborados pelo autor do mesmo, servem para fundamentação, comprovação e ilustração. É um elemento opcional)**

ANEXOS

ANEXO A - Demonstrativo de freqüência diária ago./set. 2001

ANEXO B - Demonstrativo de freqüência diária jan./dez. 2002