

PONTIFÍCIA UNIVERSIDADE CATÓLICA DO RIO GRANDE DO SUL
ESCOLA POLITÉCNICA
CURSO DE BACHARELADO EM ENGENHARIA DE SOFTWARE
SISTEMAS OPERACIONAIS

RELATÓRIO MÁQUINA VIRTUAL

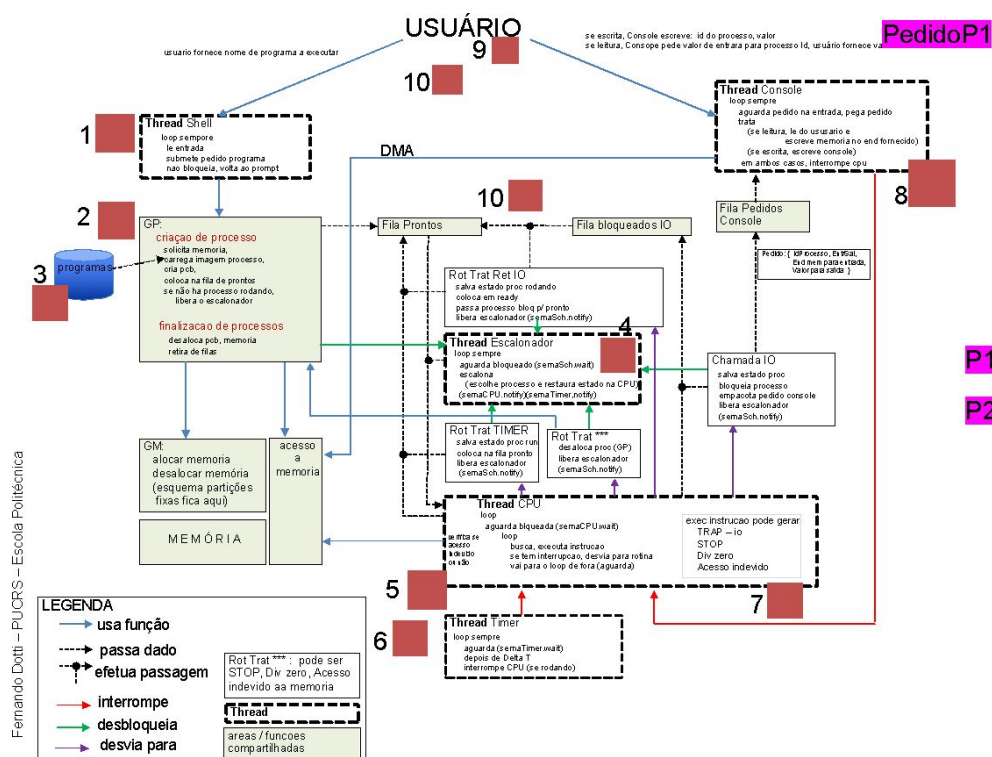
Autores: Guilherme Cavalheiro, Jamil Bannura e Marco Goedert.

Porto Alegre, Rio Grande do Sul
2020/1

1. Introdução

O trabalho final da disciplina de Sistemas Operacionais se tratava da implementação de uma Máquina Virtual, que tem como objetivo simular o que acontece dentro de um sistema operacional de verdade. A máquina virtual foi construída ao longo de todo o semestre da disciplina e este relatório descreve e detalha todo desenvolvimento do projeto na linguagem Java, assim como as experiências aprendidas e as dificuldades encontradas na implementação do trabalho proposto.

2. O QUE ESTÁ IMPLEMENTADO (OU NÃO)



O trabalho final da disciplina foi desenvolvido com base no plano de evolução sugerida pelo professor. O plano, que ajudou muito o processo de desenvolvimento, está dividida em 5 partes: a fase inicial de reaproveitamento de código e outras quatro fases enumeradas de 1 a 4. Pode ser encontrado no quadro abaixo o que foi implementado (ou não) em relação ao plano original:

Quadro 1:

| | | |
|----|------------------------|--|
| OK | Fase de Aproveitamento | <ul style="list-style-type: none"> ✓ Mem, GM (aloca e desaloca), ✓ CPU – sincronizar funcionamento ✓ GP (cria e desaloca processos) |
|----|------------------------|--|

| | | |
|-----|--------|--|
| OK | Fase 1 | <ul style="list-style-type: none"> ✓ Shell – como uma thread ✓ CPU – como uma thread ✓ Escalonador - como uma thread ✓ Fila de prontos ✓ Timer – como uma thread |
| OK | Fase 2 | <ul style="list-style-type: none"> ✓ Introduzir noção de interrupção ✓ Rotina de tratamento de interrupção ✓ Interrupção: durante a execução STOP, DivZero, Acesso indevido <ul style="list-style-type: none"> ✓ Em todas estas – pausa processo e retira ele do sistema, escalona outro |
| ≡OK | Fase 3 | <ul style="list-style-type: none"> ✓ IO só com saída. <ul style="list-style-type: none"> ✓ Só Out: processo pode fazer OUT - ocorre TRAP (instrução definida) ✓ TRAP: segue mesmo fluxo que tratamento de interrupcao. <ul style="list-style-type: none"> ✓ Desvia para uma rotina específica ✓ Fila de Bloqueados <ul style="list-style-type: none"> ○ Fila da console ○ Thread console: Lê o pedido de OUT, escreve na tela (int), manda interrupção para CPU ○ Desvia para rotina de Rot Trat Ret IO – tira processo de bloqueado e coloca em ready. |
| !OK | Fase 4 | <ul style="list-style-type: none"> ○ Entrada: mesma ideia, só que o console lê o valor e guarda na memória, área do processo requerente. |

3. PROBLEMAS ENFRENTADOS

Nosso maior problema enfrentado durante o trabalho sem dúvidas foi ter de adaptar o código da VM antiga (entrega 2) para o modelo apresentado no diagrama pelo professor, pois é complicado de realizar testes e debugs enquanto a adaptação não se finalizada. Após a adaptação, testes foram realizados e a VM executou de forma apropriada, permitindo a continuação do desenvolvimento. Outro problema que dificultou o desenvolvimento foi a perda de um colega de grupo de forma inesperada.

4. BREVE EXPLANAÇÃO DA ESTRUTURA

A estrutura é dividida em Memória (Gerenciador de Memória), Processador (Gerenciador de processos, CPU, PCBs, Fila de prontos, Fila de bloqueados, Escalonador e Timer), Programas (todos os 4 programas criados no início do

semestre) e o Shell. A ideia principal do grupo foi a de desenvolver o trabalho final de forma mais próxima possível do diagrama e do plano de evolução disponibilizado pelo professor.

5. PASSO A PASSO PARA EXECUÇÃO

A thread “main” se encontra no Shell. Após iniciar a execução, um menu de programas disponíveis aparece para o usuário e a thread aguarda o input inicial do usuário. Em vez do usuário digitar o nome dos programas, uma enumeração foi realizada, simplificando a experiência do usuário. É possível enviar múltiplos inputs ao mesmo tempo, por exemplo “1 2 3”, sem precisar enviar 1 a 1. Após, o resultado será exibido no terminal de acordo com o fim da execução dos programas pela CPU. É sugerido não enviar mais do que 8 inputs ao mesmo tempo, visto que a VM não está totalmente finalizada e exceções podem surgir do sistema. Para encerrar a execução, basta enviar a opção “-1” ao shell. Para exibir novamente o menu, basta enviar “0”.