

Relatório final da cadeira de Sistemas Operacionais

Aluno: Raul Steinmetz

Relatórios de Execução

Os trabalhos estão em formato zip no e-mail.

Trabalho 0: Trabalho foi de grande importância para o entendimento do funcionamento do projeto geral de sistema operacional. O dispositivo RANDOM foi implementado, funcionando completamente e livre de bugs (até onde eu sei). Segue um print de parte da execução.

```
PC=0000 A=000000 X=000000 10 SOMA 2
PC=0002 A=000000 X=000000 00 NOP
PC=0003 A=000000 X=000000 07 MVAX
PC=0004 A=000000 X=000000 23 LE 3
PC=0006 A=000083 X=000000 24 ESCR 0
[SAÍDA: 83]
PC=0008 A=000083 X=000000 09 INCX
PC=0009 A=000083 X=000001 08 MVXA
PC=0010 A=000001 X=000001 11 SUB 0
PC=0012 A=-00009 X=000001 18 DESVNZ 4
PC=0004 A=-00009 X=000001 23 LE 3
PC=0006 A=000086 X=000001 24 ESCR 0
[SAÍDA: 86]
PC=0008 A=000086 X=000001 09 INCX
PC=0009 A=000086 X=000002 08 MVXA
PC=0010 A=000002 X=000002 11 SUB 0
PC=0012 A=-00008 X=000002 18 DESVNZ 4
PC=0004 A=-00008 X=000002 23 LE 3
PC=0006 A=000077 X=000002 24 ESCR 0
[SAÍDA: 77]
PC=0008 A=000077 X=000002 09 INCX
PC=0009 A=000077 X=000003 08 MVXA
PC=0010 A=000003 X=000003 11 SUB 0
PC=0012 A=-00007 X=000003 18 DESVNZ 4
PC=0004 A=-00007 X=000003 23 LE 3
PC=0006 A=000015 X=000003 24 ESCR 0
[SAÍDA: 15]
```

Trabalho 1: No trabalho 1 foi implementado no SO o conceito de processos, tabela de processos, interrupções por escrita e leitura e trocas entre processos. Funciona corretamente e livre de bugs (até onde eu sei). Foi importante para um maior entendimento conceitual e a nível de código do que é um processo e como o Sistema Operacional trata trocas de processos. Alguns prints da execução em seguida:

```

Sa
Ea
Sb
Eb
Sc
Ec
Sd
Ed
Se
Ee
Sf
Ef
Sg
Eg
Sh
Eh
PC=0004 A=000003 X=000000 18 DESVNZ 0
C [OK]
PROCESSO 1 CRIADO
PROCESSO 2 CRIADO
PROCESSO 0 FINALIZADO
PROGRAMA 1 SENDO EXECUTADO!
PROGRAMA 2 SENDO EXECUTADO!
P=para C=continua S=passo Lt=LM-CM-* Zt=zera Etn=entra

```

```

Sa      1      2      3      4 5
Ea
Sb
Eb
Sc
Ec
Sd
Ed
Se
Ee
Sf
Ef
Sg
Eg
Sh
Eh
PC=0016 A=000003 X=000006 18 DESVNZ 12
PROCESSO 2 CRIADO
PROCESSO 0 FINALIZADO
PROGRAMA 1 SENDO EXECUTADO!
PROGRAMA 2 SENDO EXECUTADO!
ea10 [OK]
PROGRAMA 1 SENDO EXECUTADO!
sc
P=para C=continua S=passo Lt=LM-CM-* Zt=zera Etn=entra

```

```

Sa      1      2      3      4 5
Ea
Sb      1      2      3      4 5
Eb
Sc
Ec
Sd
Ed
Se
Ee
Sf
Ef
Sg
Eg
Sh
Eh
PC=0016 A=000003 X=000006 18 DESVNZ 12
PROGRAMA 1 SENDO EXECUTADO!
PROGRAMA 2 SENDO EXECUTADO!
ea10 [OK]
PROGRAMA 1 SENDO EXECUTADO!
eb10 [OK]
PROGRAMA 2 SENDO EXECUTADO!
P=para C=continua S=passo Lt=LM-CM-* Zt=zera Etn=entra

```

```

Sa      1      2      3      4      5
Ea
Sb      6      7      8      9     10
Eb
Sc
Ec
Sd
Ed
Se
Ee
Sf
Ef
Sg
Eg
Sh
Eh
PC=0028 A=000001 X=000011 25 SISOP 3
PROGRAMA 1 SENDO EXECUTADO!
eb10 [OK]
PROGRAMA 2 SENDO EXECUTADO!
zb [OK]
PROGRAMA 2 SENDO EXECUTADO!
PROCESSO 2 FINALIZADO
P=para C=continua S=passo Lt=lM-CM-* Zt=zera Etn=entra

```

Trabalho 2: Neste trabalho, foram implementados dois escalonadores: O Round Robin e o Processo mais Curto, além de implementar preempções por tempo de CPU (relógio). Também foram calculadas métricas para diferentes tempos máximos de CPU (quantum) em ambos os escalonadores. Esse trabalho foi especialmente útil para esclarecer as dúvidas sobre como as preempções funcionam, além de entender como escalonadores operam. Print da execução a seguir:

```

Sa      6      7      8      9     10
Ea
Sb      6      7      8      9     10
Eb
Sc
Ec
Sd
Ed
Se
Ee
Sf
Ef
Sg
Eg
Sh
Eh
PC=0028 A=000001 X=000011 25 SISOP 3
Tempo total: 297      Uso da Cpu: 264      Interrupcoes: 340
P      R      B      E      esp      mr      b      pre
0      4      0      4      0      0,00    0      0
1      204    4      130    70      4,00    1      3
2      293    90      131    72      90,00   1      3
Fim da execucao      Relogio: 297
digite ENTER para sair P=para C=continua S=passo Lt=lM-CM-* Zt=zera Etn=entra

```

Logs dos diversos testes a seguir, com mais detalhes no pdf anexado no zip:

Processo mais Curto

Quantum 12 - entradas digitadas antes da execução (ea5 eb5)

Tempo total: 141 Uso da Cpu: 140 Interrupcoes: 178

P	R	B	E	esp	mr	b	pre
0	4	0	4	0	0,00	0	0
1	131	0	68	63	0,00	0	5
2	137	0	68	69	0,00	0	5

Fim da execucao Relogio: 141

Quantum 12 - entradas digitadas sobre demanda (ea10 eb10 za zb)

Tempo total: 409 Uso da Cpu: 288 Interrupcoes: 484

P	R	B	E	esp	mr	b	pre
0	4	0	4	0	0,00	0	0
1	329	182	144	3	91,00	2	10
2	405	257	144	4	128,00	2	10

Fim da execucao Relogio: 409

Quantum 30 - entradas digitadas antes da execução (ea5 eb5)

Tempo total: 123 Uso da Cpu: 122 Interrupcoes: 144

P	R	B	E	esp	mr	b	pre
0	4	0	4	0	0,00	0	0
1	92	0	59	33	0,00	0	1
2	119	0	59	60	0,00	0	1

Fim da execucao Relogio: 123

Quantum 30 - entradas digitadas sobre demanda (ea10 eb10 za zb)

Tempo total: 378 Uso da Cpu: 270 Interrupcoes: 425

P	R	B	E	esp	mr	b	pre
0	4	0	4	0	0,00	0	0
1	308	169	134	5	84,00	2	3
2	374	235	135	4	117,00	2	3

Fim da execucao Relogio: 378

Round Robin

Quantum 12 - entradas digitadas antes da execução (ea5 eb5)

Tempo total: 141 Uso da Cpu: 140 Interrupcoes: 178

P	R	B	E	esp	mr	b	pre
0	4	0	4	0	0,00	0	0
1	131	0	68	63	0,00	0	5
2	137	0	68	69	0,00	0	5

Fim da execucao Relogio: 141

Quantum 12 - entradas digitadas sobre demanda (ea10 eb10 za zb)

Tempo total: 485 Uso da Cpu: 288 Interrupcoes: 560

P	R	B	E	esp	mr	b	pre
0	4	0	4	0	0,00	0	0
1	408	261	144	3	130,00	2	10
2	481	333	144	4	166,00	2	10

Fim da execucao Relogio: 485

Quantum 30 - entradas digitadas antes da execução (ea5 eb5)

Tempo total: 123 Uso da Cpu: 122 Interrupcoes: 144

P	R	B	E	esp	mr	b	pre
0	4	0	4	0	0,00	0	0
1	92	0	59	33	0,00	0	1
2	119	0	59	60	0,00	0	1

Fim da execucao Relogio: 123

Quantum 30 - entradas digitadas sobre demanda (ea10 eb10 za zb)

Tempo total: 384 Uso da Cpu: 270 Interrupcoes: 431

P	R	B	E	esp	mr	b	pre
0	4	0	4	0	0,00	0	0
1	319	181	135	3	90,00	2	3
2	380	241	135	4	120,00	2	3

Fim da execucao Relogio: 384

Trabalho 3 (Parte 1): Foi implementada a memória principal do SO grande o suficiente para que caibam todos os programas, evitando a necessidade de ter uma memória por programa. Essa parte do trabalho ajudou no entendimento da mmu e a estrutura de uma tabela de páginas. Prints da execução a seguir:

```
Sa
Ea      10
Sb
Eb      10
Sc
Ec      3
Sd
Ed
Se
Ee
Sf
Ef
Sg
Eg
Sh
Eh
PC=0002 A=000001 X=000000 25 SISOP 4

QUADROS 2
ea10 [OK]
eb10 [OK]
ec3 [OK]
```

P=para C=continua S=passo Lt=LM-CM-* Zt=zera Etn=entra

```
Sa      1      2      3      4      5      6      7      8      9
Ea
Sb      1      2      3      4      5      6      7      8      9
Eb
Sc      590     73     84     73     84     590
Ec
Sd
Ed
Se
Ee
Sf
Ef
Sg
Eg
Sh
Eh
PC=0012 A=000000 X=000590 25 SISOP 3
PROCESSO 3 CRIADO
PROCESSO 0 FINALIZADO
PROGRAMA 1 SENDO EXECUTADO! PC = 0
PROGRAMA 2 SENDO EXECUTADO! PC = 0
PROGRAMA 3 SENDO EXECUTADO! PC = 0
PROCESSO 3 FINALIZADO
P=para C=continua S=passo Lt=lM-CM-* Zt=zera Etn=entra

Sa      10
Ea
Sb      10
Eb
Sc      590     73     84     73     84     590
Ec
Sd
Ed
Se
Ee
Sf
Ef
Sg
Eg
Sh
Eh
PC=0028 A=000001 X=000011 08 MVXA
0        6        0        6        0        0,00    0        0
1       2185     2068     112     5        2068,00  1        0
2       2243     2027     112    104       2027,00  1        0
3        496      0       293    203        0,00    0        0
Fim da execuM-CM-'M-CM-#o.
relM-CM-3gio: 2247
digite ENTER para sair P=para C=continua S=passo Lt=lM-CM-* Zt=zera Etn=entra
```

Trabalho 3 (Parte 2): Por mais que eu tenha compreendido a nível de conceito de tabela de páginas, mmu, page faults, escalonadores e cache, eu fiquei até o limite do tempo tentando implementar mas não cheguei em lugar nenhum.

Auto-Avaliação

No início, gostaria de escrever sobre minha experiência de aprendizado durante a disciplina. Embora eu não tenha sido frequentemente presente nas aulas, eu me esforcei para estudar

[illegible][illegible]

ATIVIDADES DE PROCESSO

- 1. Execução: movimento lógico (A) no momento.
- 2. Algoritmo: sequência de operações que serão realizadas uma a uma.
- 3. Primeiro: uma sequência, porém que poderá sofrer uma interrupção, mesmo quando a complexidade do processo.

EM EXECUÇÃO

1. Primeiro bloqueio após entrada iniciada
2. Realizada relação entre processamento e pronto.
3. Realizada relação entre processamento e pronto.
4. Entrada bloqueada de processamento

E SCAONADOR

- ↳ Traça os dados
- ↳ Algoritmo que realiza qual qual o processo no momento
- ↳ Diversas técnicas
- ↳ SCAONADOR e QUALQUER DADO DO (MATERIAIS PRONTO)

ORGANIZAÇÃO DE CONTEÚDO

- ↳ O nível lógico de dados que qual qual o conteúdo do conteúdo em cada um dos processos que qual qual o conteúdo no CPU
- ↳ Conteúdo
- ↳ Algoritmo de organização, organização de conteúdo de dados
- ↳ Algoritmo de organização de dados (qual qual o conteúdo de dados)

SOLUÇÕES DE EXCLUSÃO MÚTUA

- ↳ ESPERA OUTRA
 - ↳ ACCEPT/WORK
 - ↳ DECREMENT
 - ↳ INQUIRIES
 - ↳ TROCA DE MENSAGEM.
- ↳ EVITAR O DEADLOCK DE RECURSOS
 - ↳ (bus waiting) \rightarrow 4 formas
- ↳ ESPERA OCUPADA
 - ↳ DESABILITAR INTERRUPTORES
 - ↳ DESABILITAM
 - ↳ O tempo de uso do CPU
 - ↳ (no tempo que um núcleo recebe, desabilita todos os interruptores do núcleo, CPU não chama mais processamento, se o processador não recebe mais R.C., daí desabilita.)
 - ↳ Se houver um sistema não cooperativo, não se faz o mesmo.
 - ↳ Se houver um sistema não cooperativo de uma só vez, não desabilita o acesso ao compartilhamento de recursos, pois desabilita que outros acessem o R.C.
- ↳ VARIÁVEIS LÓGICAS
 - ↳ Sistema que recebe o recurso R.C., atualiza um valor a uma variável compartilhada globalmente.
 - ↳ Se a variável LÓGICA \neq 0, o processamento não usa o R.C.
 - ↳ Se $x = 1$, recebe um processo que R.C.
 - ↳ Podemos ter que x não recebe de outro \rightarrow $x = 1$ e não recebe.
 - ↳ O que fazemos um processo P_1 que tem o R.C. e não recebe de outro \rightarrow P_1 não recebe mais o R.C. e não recebe mais o R.C. e não recebe mais o R.C.

[illegible][illegible]

Dos seguintes tópicos citados na especificação da Entrega Final na página da disciplina, acredito que obtive um bom entendimento de todos durante a disciplina, com uma pequena exceção na parte de ideias de implementação de sistemas de arquivos, que vi por cima (não sei se eu deveria saber as especificidades das principais, mas não sei).

- gerência de processos (programa x processo; modo supervisor + usuário; interrupções; quando o estado de um processo é trocado; exemplos de escalonador em cada classificação de SO)
- comunicação entre processos/threads (condição de corrida; região crítica; exclusão mútua; mecanismos de sincronização)
- gerência de memória (proteção + relocação; memória virtual; algoritmos de substituição de páginas)
- gerência de E/S (dispositivo + controladores; formas de comunicação; camadas de sw de e/s)
- sistemas de arquivos (arquivos + diretórios; principais idéias de implementação)
- deadlocks (recursos; condições para um deadlock; estratégias para lidar com deadlocks)

Levando todos os argumentos citados acima em consideração e considerando que o peso da aprendizagem do conteúdo da disciplina seria igual ao peso dos trabalhos, na parte de aprendizado eu me avaliaria com 4.75/5, levando em conta que todos os tópicos têm o mesmo peso e que não me aprofundei em implementações de sistema de arquivos apenas.

Na parte dos trabalhos, acredito que por eu ter implementado o T0, T1, T2 e T3 parte 1, com tudo o que foi pedido e funcionando, eu me avaliaria com a nota 3/5, por não ter finalizado a parte 2 do T3, além de claramente pecar na organização do código, pois me programei mal e acabei fazendo todos os trabalhos nas últimas semanas, junto com todos os outros trabalhos e provas de outras disciplinas, tal fato foi um erro completamente meu.

No geral, sem saber com acurácia o peso de cada trabalho, e considerando meu aprendizado da disciplina, eu me avaliaria com $(4.75 + 3.0)$, **7.75/10**, mesmo entendendo que possivelmente minha noção da importância de cada atividade exercida na disciplina pode estar errada, reduzindo ou aumentando a nota.