Trabalho IV – Gerência de E/S & Deadlocks

Vinícius Takeo Friedrich Kuwaki

17 de Setembro de 2020

Exercício 1

A partir do conjunto A = $\{59, 603, 53, 700, 1785, 474, 1446, 2, 1690, 1368\}$ e afirmando que o cabeçote encontra-se no cilindro 210, as figuras 1, 2 e 3 a seguir representam o escalonamento realizado pelos algoritmos FCFS, SSF e Elevador, respectivamente. O resultado final é apresentado na tabela 1.

FCFS

A matriz de distâncias apresentada na figura 1 foi gerada utilizando um pequeno algoritmo em C (q1.c), que considerou o valor absoluto da subtração entre cada linha e coluna da matriz, esta composta pelas requisições. O algoritmo então gerou um arquivo .csv, cujo seu conteúdo foi copiado e colado em uma planilha do Google Sheets. O resultado obtido na planilha é a própria figura 1. A célula descada em roxo representa a distância total, essa obtida pelo somatório dos valores das células em verde. Como o algoritmo atende as requisições na ordem em que chegam, as colunas acima da diagonal principal, foram pintadas de verde, representando um caminho que abrange todas as requisições. A distância total de tal caminho é de 8714, é possível ver a lista obtida pelo algoritmo na tabela 1, assim como a distância total.

FCSF											
8714	210	59	603	53	700	1875	474	1446	2	1690	1368
210		151	393	157	490	1575	264	1236	208	1480	1158
59			544	6	641	1726	415	1387	57	1631	1309
603		544		550	97	1182	129	843	601	1087	765
53		6	550		647	1732	421	1393	51	1637	1315
700		641	97	647		1085	226	746	698	990	668
1875		1726	1182	1732	1085		1311	339	1783	95	417
474		415	129	421	226	1311		972	472	1216	894
1446		1387	843	1393	746	339	972		1444	244	78
2		57	601	51	698	1783	472	1444		1688	1366
1690		1631	1087	1637	990	95	1216	244	1688		322
1368		1309	765	1315	668	417	894	78	1366	322	

Figura 1: Aplicação do Algoritmo FCFS no conjunto A.

O tempo total de leitura é então calculado pelas expressões a seguir:

$$t_{seek_{FCFS}} = 8714 * 0.5 * 10^{-3} = 4,357s \tag{1}$$

$$t_{latenciaFCFS} = 10 * \frac{1}{2 * 100} = 0,050s \tag{2}$$

$$t_{transferenciaFCFS} = 10 * \frac{8000}{100 * 65536} = 0,012s \tag{3}$$

$$t_{acessoFCFS} = 4,357 + 0,050 + 0,012 = 4,420s \tag{4}$$

SSF

A matriz de distâncias apresentada na figura 2 também foi gerada pelo algoritmo (q1.c). Entretanto, as requisições não são atendidas na ordem em que chegam, tal como o FCFS o faz. As células em roxo e verde tem o mesmo significado da seção anterior, entretanto o cálculo das células em verde é diferente. O algoritmo começa buscando na primeira linha da matriz, qual a menor distância. A partir disso ela encontra a célula ij de menor valor e repete o mesmo processo nessa linha j, até que todas as requisições tenham sido atendida, isto é, cada linha e coluna possua apenas uma célula em verde. O resultado final pode ser visto na tabela 1.

SSF											
1991	210	59	603	53	700	1875	474	1446	2	1690	1368
210		151	393	157	490	1575	264	1236	208	1480	1158
59			544	6	641	1726	415	1387	57	1631	1309
603		544		550	97	1182	129	843	601	1087	765
53		6	550		647	1732	421	1393	51	1637	1315
700		641	97	647		1085	226	746	698	990	668
1875		1726	1182	1732	1085		1311	339	1783	95	417
474		415	129	421	226	1311		972	472	1216	894
1446		1387	843	1393	746	339	972		1444	244	78
2		57	601	51	698	1783	472	1444		1688	1366
1690		1631	1087	1637	990	95	1216	244	1688		322
1368		1309	765	1315	668	417	894	78	1366	322	

Figura 2: Aplicação do Algoritmo SSF no conjunto A.

O tempo total de leitura é então calculado pelas expressões a seguir:

$$t_{seek_{SSF}} = 1991 * 0.5 * 10^{-3} = 0,995s \tag{5}$$

$$t_{latenciaSSF} = 10 * \frac{1}{2 * 100} = 0,050s \tag{6}$$

$$t_{transferenciaSSF} = 10 * \frac{8000}{100 * 65536} = 0,012s \tag{7}$$

$$t_{acessoSSF} = 0,995 + 0,050 + 0,012 = 1,057s \tag{8}$$

Elevador

Já matriz de distâncias do algoritmo do elevador, apresentada na figura 3, foi gerada utilizando o algoritmo q1Ordenar.c. Nela, as requisições foram ordenadas crescentemente pelo algoritmo. O algoritmo difere um pouco do SSF. A partir do cilindro 210 (isto é, considera apenas os cilindros que possuem valores maiores que 210) o algoritmo procura pela menor célula ij na linha, nunca considerando as colunas em que j é menor que i. Quando todas as requisições até 210 forem atendidas, o algoritmo volta para a primeira linha e continua sempre movendo-se crescentemente (isto é, para a direita) e tal como no SSF, apenas uma célular em cada linha e coluna deve ser pintada de verde, gerando o resultado apresentado na tabela 1.

Elevador

3358	2	53	59	210	474	603	700	1368	1446	1690	1785
2		51	57		472	601	698	1366	1444	1688	1783
53	51		6		421	550	647	1315	1393	1637	1732
59	57	6			415	544	641	1309	1387	1631	1726
210					264	393	490	1158	1236	1480	1575
474	472	421	415			129	226	894	972	1216	1311
603	601	550	544		129		97	765	843	1087	1182
700	698	647	641		226	97		668	746	990	1085
1368	1366	1315	1309		894	765	668		78	322	417
1446	1444	1393	1387		972	843	746	78		244	339
1690	1688	1637	1631		1216	1087	990	322	244		95
1785	1783	1732	1726		1311	1182	1085	417	339	95	

Figura 3: Aplicação do Algoritmo do Elevador no conjunto A.

O tempo total de leitura é então calculado pelas expressões a seguir:

$$t_{seek_{Elevador}} = 3358 * 0.5 * 10^{-3} = 1,679 \tag{9}$$

$$t_{latenciaElevador} = 10 * \frac{1}{2 * 100} = 0,05s \tag{10}$$

$$t_{transferenciaElevador} = 10 * \frac{8000}{100 * 65536} = 0,012s \tag{11}$$

$$t_{acessoElevador} = 1,679 + 0,05 + 0,012 = 1,741s \tag{12}$$

Exercício 2

Para três processos A,B e C e quatro recursos W,X,Y e Z, uma sequência de execução que gera um deadlock envolvendo os três processos é apresentada nas figuras 4 a 6, representada através de um grafo de alocações de recursos. A mesma sequência de execução que gera o deadlock é apresentada na tabela 2.

Já na figura 7 é possível ver uma sequência de execução que não gera deadlock.

	FCSF	SSF	Elevador
Sequência de Atendimentos	59	59	474
	603	53	603
	53	2	700
	700	474	1368
	1875	603	1446
	474	700	1690
	1446	1368	1785
	2	1446	59
	1690	1690	53
	1368	1875	2
Distância Total	8714	1991	3358

Tabela 1: Sequência de atendimentos geradas pelos algoritmos FCSF, SSF e Elevador e suas respectivas distâncias totais.

Sequência	Comando
$B1^+$	B requisita Y
$C1^+$	C requisita W
$C2^+$	C requisita Z
$A1^-$	A requisita W
$B2^-$	B requisita Z
$C3^-$	C requisita Y
$B3^-$	B usa Y+Z

Tabela 2: Sequência de execuções que levam a um deadlock entre os três processos A, B e C.

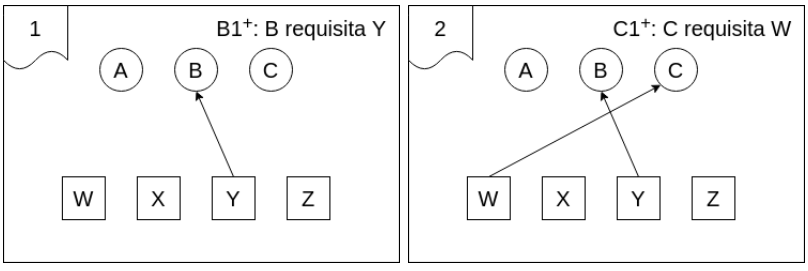

Figura 4: Na esquerda, B consegue alocar o recurso Y, já na direita, C também consegue alocar o recurso W.

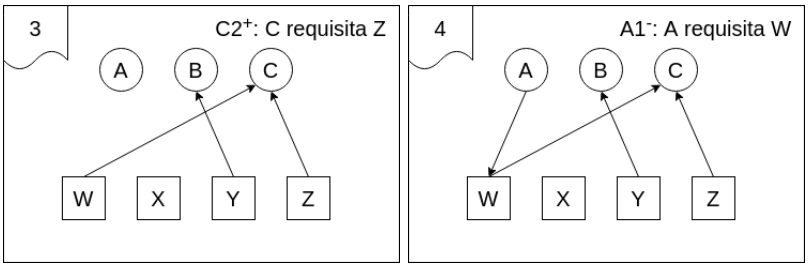

Figura 5: Na esquerda C consegue alocar o recurso Z e na direita A não consegue alocar o recurso W, que está em posse de C.

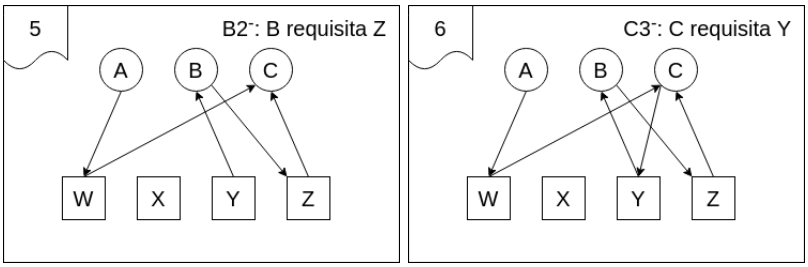

Figura 6: Na esquerda B falha em alocar o recurso Z, pois ele está com C. Enquanto na direita C também falha em alocar Y.

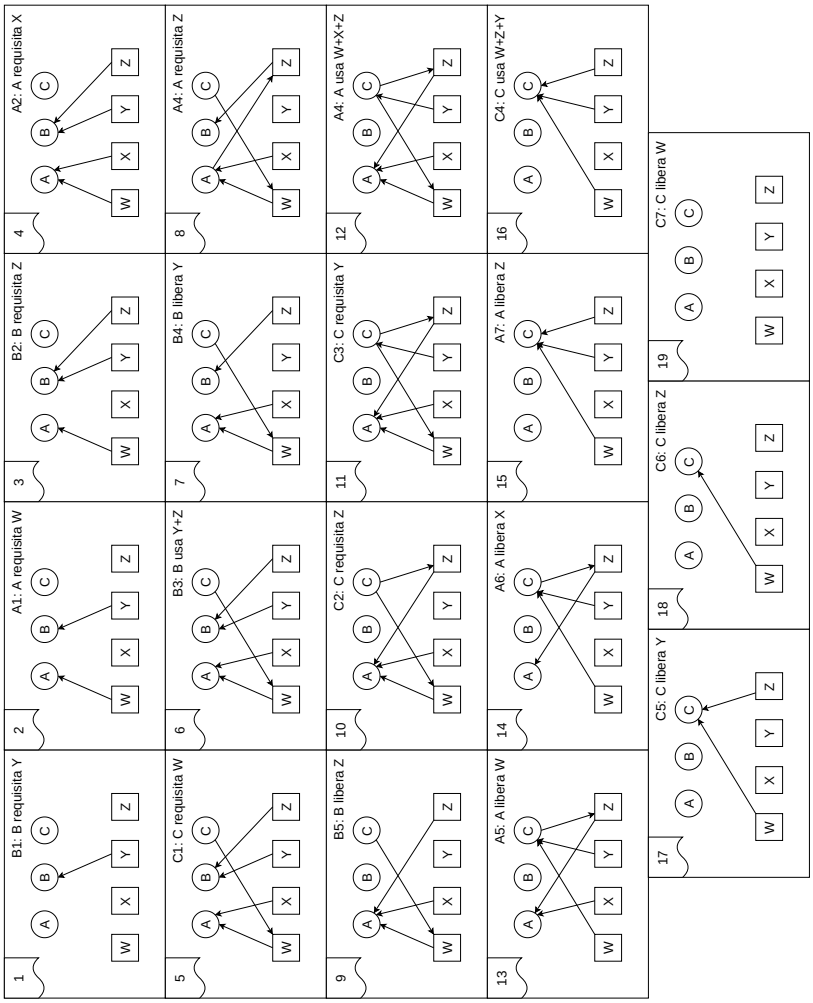

Figura 7: Modificando a ordem dos comandos de forma a não gerar um deadlock.