

Sistemas Operacionais – 2016/1 – Profa. Roberta Lima Gomes

Resultados – Trabalho 2

		NOTA
2014100468 2014100498 2014100491	[-0,5] a função "synchronized void addMsg2Impar(String msg)" não permite que 2 threads insiram msgs simultaneamente no mesmo buffer. A solução seria colocar sychronized só na R.C., e nela apenas copiar o valor do índice e incrementá-lo. [+0,2] EXTRA: interface de visualização	9,7
2014100465 2014100637	[-0,5] Atraso [-0,3] Mal comentado [-0,2] No código java, a leitura do pipe não precisa ficar dentro do "synchronized"... a sincronização é inerente ao próprio pipe [-1,0] No código java, não foram criados os buffer corretamente, e não há paralelismo efetivo na escrita (caso tivesse buffers) [-0,5] Sem readme/instruções de compilação/execução	7,5
2013101883 2013101034 2011100767	[-0,3] Atraso [-0,3] No código abaixo, a condição de parada está fora da R.C.: <pre> While (num <= 999) { pthread_mutex_lock(&TrabFIFO_mutex); sprintf(msg, "%d-MSG-%03d\n", id_thread, num++); pthread_mutex_unlock(&TrabFIFO_mutex); /*ESCREVE NO PIPE*/ write(fd, msg, strlen(msg)); sleep(1); } </pre> ... Com isso, mais de uma thread poderia chegar a conclusão de que "num == 999" entrando em seguida na região crítica e gerando msgs com ids maiores do que 999, por exemplo: 3-MSG-999 5-MSG-1000 4-MSG-1002 1-MSG-1001 2-MSG-1003 [-1,0] Vocês não usaram synchronized para garantir exculsão mútua na inserção no buffer... não há garantias de comportamento quando duas threads tentam simultaneamente fazer um add() em um ArrayList!	8,4
2013101857 2013101847 2013101226	[-2,0] Erro de execução no código Java: Exception in thread "Thread-1" Exception in thread "Thread-0" java.lang.ArrayIndexOutOfBoundsException: 2 at NamedPipe.isEven(NamedPipe.java:27) at NamedPipe.run(NamedPipe.java:41) java.lang.NumberFormatException: For input string: "MSG" at java.lang.NumberFormatException.forInputString(NumberFormatException.java:65) at java.lang.Integer.parseInt(Integer.java:492) at java.lang.Integer.parseInt(Integer.java:527) at NamedPipe.isEven(NamedPipe.java:27) at NamedPipe.run(NamedPipe.java:41) Exception in thread "Thread-3" java.lang.ArrayIndexOutOfBoundsException: 2 at NamedPipe.isEven(NamedPipe.java:27) at NamedPipe.run(NamedPipe.java:51) Exception in thread "Thread-2" java.lang.ArrayIndexOutOfBoundsException: 2 at NamedPipe.isEven(NamedPipe.java:27) at NamedPipe.run(NamedPipe.java:51) [-0,2] Vocês colocaram "synchronized" para implementar exculsão mútua na leitura do pipe, o que não é necessário uma vez que o próprio pipe já garante essa sincronização. [-0,5] Vocês implementaram a exculsão mútua de forma que duas threads não conseguem escrever ao mesmo tempo nos dois buffers, oi ainda no mesmo buffer. [-0,3] Mal comentado	7,0
2013100847	[-0,1] Threads já compartilha memória por princípio... você não deve usar memória compartilhada (que servae para compartilhar memória entre processos [-0,1] "int fd = open("tmpNamedPipe", O_WRONLY);" dentro do while(1)... conceitualmente, bastaria abrir 1 vez p/ o processo inteiro já que a tabela de descritores é única para todas as threads! [-1,0] Jvc definiu "synchronized void EscreveParaBuffer(InfoThread info, String msg)", com isso duas threads NÃO conseguem escrever ao mesmo tempo nos buffer par e ímpar, ou ainda ao mesmo tempo só no par ou só no ímpar.	8,2
2014100470 2014100496 2014100488	[-4,0] Não compila [-2,0] Não implementou exculsão mútua no código C [-2,0] Não implementou exculsão mútua no código Java	2,0
2013101609 2013101610 2013101595	[-0,5] a função "synchronized static void insert_even(..." não permite que 2 threads insiram msgs simultaneamente no mesmo buffer. A solução seria colocar sychronized só na R.C., e nela apenas copiar o valor do índice e incrementá-lo.	9,5
2014100486 2014102172	OK!	10,0