Estrutura de Dados – 2017/1

Trabalho Prático

Uma Fábrica de Rolamentos atende a pedidos por meio de processos envolvendo um conjunto de máquinas. A fábrica é capaz de produzir rolamentos de quatro tipos: **cilíndrico**, **cônico**, **esférico de aço** ou **esférico de titânio**. Para cada tipo de rolamento, há uma prioridade e um processo ser seguido (envolvendo a ordem e o tempo nas máquinas).

Os pedidos possuem as seguintes prioridades: cilíndrico: 1 (mínima); cônico: 2; esféricos: 3 (máxima). Os pedidos de rolamentos esféricos são 90% das vezes de aço e somente 10% das vezes de titânio.

A empresa conta hoje com quatro máquinas: **1 Fresa**, **1 Mandril** e **2 Tornos**. Os pedidos chegam às máquinas por meio de 3 Filas de Prioridade, uma para cada tipo de equipamento.

A fábrica consegue determinar antecipadamente quando o próximo pedido de cada tipo de rolamento vai chegar (função `chegadaPedido(param)`) para se preparar melhor para a sua produção. No momento em que um pedido chega, ele é encaminhado para a fila correspondente à primeira máquina de seu processo. Após percorrer a fila (considerando sua prioridade), o pedido é enviado para a máquina, que exige um tempo de processamento (função genérica `tempoMaquina()`). Após ser processado na primeira máquina, o pedido segue para a fila da próxima máquina, e assim por diante, conforme seu processo. Ao ser processado pela última máquina, o pedido é considerado entregue e seu tempo total de processamento é contabilizado.

A tabela a seguir apresenta os parâmetros do processo. Para cada rolamento, são mostrados: a sua prioridade, o parâmetro de tempo de chegada do pedido, e os equipamentos utilizados em sua fabricação, na ordem, com os respectivos parâmetros de tempo de processamento.

Rolamento	**Prior.**	**Chegada Pedido**	**Equipamentos do Processo**				
			1º	**2º**	**3º**	**4º**	**5º**
CILÍNDRICO	1	21.5	TORNO 0.8	FRESA 0.5	TORNO 0.8	MANDRIL 1.2	-
CÔNICO	2	19.1	TORNO 1.8	MANDRIL 2.1	TORNO 1.8	-	-
ESFÉRICO AÇO	3	8.0	FRESA 0.5	MANDRIL 1.4	TORNO 1.0	-	-
ESFÉRICO TITÂNIO	3	8.0	FRESA 0.6	MANDRIL 1.5	TORNO 1.6	FRESA 0.6	TORNO 1.6

Seu objetivo é desenvolver um programa em C que simule a fábrica, usando os dados definidos anteriormente e recebendo como argumento de entrada um *double* que representará o tempo de simulação total da fábrica. Este não é "tempo real", mas uma simulação, controlada pelo seu programa. A cada rolamento finalizado, o programa deve imprimir o seu tipo e o seu tempo total de processamento (incluindo esperas). Ao final, o programa deve imprimir um resumo, indicando, para cada tipo de rolamento, a quantidade produzida e o seu tempo médio de construção.

Para auxiliar a simulação: no primeiro instante a fábrica é informada da chegada dos 3 primeiros pedidos (um de cada tipo de rolamento). Sempre que um pedido chega (seu momento de chegada é atingindo), ele é colocado na fila correspondente ao seu processo e um novo pedido do mesmo tipo é informado. A fábrica somente entregará os pedidos produzidos dentro do tempo passado como argumento.

O programa deve ser bem modularizado (desenvolva bibliotecas usando .c e .h) e utilizar as estruturas de dados adequadas. Ponteiros para dados e funções devem ser utilizados conforme pertinência. Crie constantes (#define) para os parâmetros da tabela. Evite variáveis globais.

A função `chegadaPedido(param)` retorna o tempo que de demora para chegar um pedido, passando como argumento o parâmetro correto (21.5, 19.1 ou 8.0).

```
/* Gera o tempo de chegada de um novo pedido */
float chegadaPedido(float param) {
   float u=0;
   do {
      u = (float) (rand()%RAND_MAX) / RAND_MAX;
   } while ((u==0) || (u==1));
   return (float) (-param * log (u));
}
```

A função `tempoMaquina()` é genérica e deve ser implementada de forma a atender cada combinação de rolamento / máquina. *EstadiaEquipamentoRolamento* é uma das constantes definidas a partir da tabela.

```
/* Retorna o tempo de processamento em uma máquina específica. */
float tempoMaquina() {
   return 2.0 * <EstadiaEquipamentoRolamento> * rand() / (RAND_MAX + 1.0);
}
```

Bom Trabalho!