Instituto Superior de Engenharia de Lisboa

Licenciatura em Engenharia Informática e de Computadores

Sistemas Operativos, Verão de 2020/2021

Quarta série de exercícios

Para cada questão abaixo, deverá ser incluído um programa de teste que demonstre a correção da solução proposta.

1. Acrescente o necessário à biblioteca UThread para manter informação sobre o estado de cada *thread*, que poderá ser um dos seguintes: `Running`, `Ready` ou `Blocked`. Adicione a função `ut_state`, que permite consultar o estado de uma *thread* dado o seu *handle*.

2. Acrescente a noção de prioridade às *threads* da biblioteca UThread. Na criação de uma *thread* deverá ser indicada a prioridade pretendida. Em qualquer momento o *scheduler* escolherá para execução a *thread* com maior prioridade (valor numérico mais elevado). Considere as prioridades LOW(1), NORMAL(2), HIGH(3) e IDLE(0). A prioridade IDLE será usada exclusivamente pela *thread* que executa `ut_run`.

3. Implemente o objeto de sincronização `CyclicBarrier`, com o parâmetro de construção `parties`, que será sempre um valor inteiro positivo. O objeto tem apenas uma operação de sincronização, `await`, com dois comportamentos possíveis:

   - se o número total de *threads* bloqueadas na operação `await` do objeto corrente for **inferior** a `parties - 1`, a *thread* corrente fica bloqueada em `await`;
   - se o número total de *threads* bloqueadas na operação `await` do objeto corrente for **igual** a `parties - 1`, todas são desbloqueadas e a *thread* corrente prossegue sem se bloquear.

   Quando ocorre um desbloqueio, cada uma das *threads* retorna da função `await` com um valor inteiro distinto, entre `0` e `parties - 1`.

   Note que a implementação deverá respeitar sempre as prioridades das *threads*. Note ainda que qualquer uma das *threads* libertadas poderá voltar a invocar `await`, incluindo antes que todas as outras tenham tido oportunidade de voltar a correr.

Entrega

Descompacte o ficheiro fornecido em anexo para a directoria **`se4`** do repositório de grupo e resolva os exercícios sobre a mesma base de código (não são necessárias subdirectorias para cada alínea) . Não adicione o ficheiro zip ao repositório.

A entrega é finalizada usando a *tag* **SE4** no repositório GitHub.

ISEL, 16 de maio de 2021

Data limite de entrega: 26 de maio de 2021