

# L05 - Implementação de tarefas

Total de pontos **7/7** ?

Por favor preencha seu nome \*

Bruno Duarte

✓ Não são dados presentes em um PCB \*

1/1

- ☐ PID e PPID do processo
- ☐ Lista de áreas de memória usadas pela tarefa
- ☐ Estado da tarefa (nova, pronta, executando, suspensa, terminada, etc.)
- ☒ Lista de IRQs disponíveis no sistema ✓
- ☐ Informações de contexto do processador (valores contidos nos registradores)

✓ O que é troca de contexto? \*

1/1

- ☐ É a troca mecânica dos mecanismos do disco rígido
- ☐ É a operação de swap realizada da memória para o disco e vice-versa
- ☒ É a operação onde as informações de um processo em execução é salvo em seu TCB, na memória e, posteriormente, o TCB de outro processo é carregado da memória para o processador ✓
- ☐ É a ativação de uma sequência de flags de processamento importantes para manter o processador em estado conforme
- ☐ NDA



✓ Com relação aos processo, assinale as alternativas verdadeiras \* 1/1

☒ Um processo pode então conter várias tarefas, que compartilham esses recursos ✓

☐ Um processo possui, necessariamente, apenas uma tarefa

☒ Um processo ser visto como uma unidade de contexto ✓

☐ As tarefas são isoladas entre si pelos mecanismos de proteção providos pelo hardware (isolamento de áreas de memória, níveis de operação e chamadas de sistema)

☐ Processos não contém região de memória, sendo apenas uma entidade abstrata de alocação de recursos

☒ Os sistemas operacionais atuais geralmente associam por default uma única tarefa a cada processo ✓

✓ Processos podem solicitar ao núcleo o encerramento de outros processos, porém... \* 1/1

☐ Essa operação só é permitida para processo do usuário root

☒ Essa operação só é aplicável a processos do mesmo usuário ✓

☐ Essa operação só é aplicável a processos do mesmo grupo

☐ Essa operação só é permitida antes de uma chamada wait

☐ NDA



✓ A API POSIX Threads (ou simplesmente pthreads) implementa atualmente o modelo de gerenciamento: \*

1/1

- ☐ Modelo 1:1
- ☐ Modelo N:1
- ☐ Modelo N:M
- ☒ Nenhuma das alternativas pois a API apenas especifica a interface de uso, não sua implementação. O modelo de gerenciamento depende do SO, e não da API
- ☐ NDA



✓ Para o que serve a função fork()? \*

1/1

- ☐ O processo pai volta a rodar somente após o término do filho, ou seja, o processo pai aguarda o filho ser finalizado
- ☐ É uma chamada de sistema que termina um processo e retorna um status
- ☒ É uma chamada de sistema usada para criar um novo processo "filho", idêntico ao pai
- ☐ Altera o sistema de proteção do arquivo passado como parâmetro



✓ Numa chamada de sistema fork() que foi invocada por um processo, temos o processo pai (quem invocou o fork()) e filho, quem recebe o retorno? \*

1/1

- ☐ Somente o processo pai
- ☐ Um terceiro processo é gerado e recebe o retorno
- ☒ O processo pai, que a invocou, e o processo filho, que acabou de ser criado
- ☐ Somente o processo filho



Este conteúdo não foi criado nem aprovado pelo Google. - [Termos de Serviço](#) - [Política de Privacidade](#)

## Google Formulários

