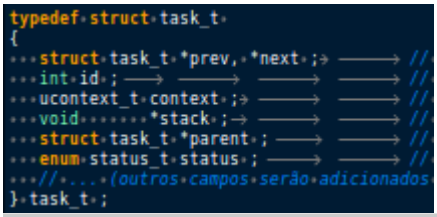

Carlos Maziero

Trocas de contexto

Implementar tarefas simultâneas dentro de um processo de usuário não é uma tarefa difícil, embora alguns detalhes de baixo nível possam assustar os iniciantes, como a manipulação de registradores para as trocas de contexto. Felizmente, algumas chamadas de sistema POSIX permitem simplificar a manipulação de contextos, eliminando as operações com registradores e tornando o código portável:

- `getcontext(&a)` : salva o contexto atual na variável `a`.
- `setcontext(&a)` : restaura um contexto salvo anteriormente na variável `a`.
- `swapcontext(&a,&b)` : salva o contexto atual em `a` e restaura o contexto salvo anteriormente em `b`.
- `makecontext(&a, …)` : ajusta alguns valores internos do contexto salvo em `a`.
- as variáveis `a` e `b` são do tipo `ucontext_t` e armazenam contextos de execução.

Mais informações sobre essas funções podem ser obtidas em suas respectivas páginas de manual (`man getcontext`, etc.).

Estude o código presente no arquivo contexts.c, execute-o e explique seu funcionamento.

Elabore um relatório (no formato correto) cobrindo pelo menos os seguintes pontos:

1. Explicar o objetivo e os parâmetros de cada uma das quatro funções acima.
2. Explicar o significado dos campos da estrutura `ucontext_t` que foram utilizados no código.
3. Explicar cada linha do código de `pingpong.c` que chame uma dessas funções ou que manipule estruturas do tipo `ucontext_t`.
4. Desenhar o diagrama de tempo da execução.

Outras informações

- Duração estimada: 2 horas.

so/trocas_de_contexto.txt · Última modificação: 2015/03/27 14:06 por maziero