Relatório projeto 1
Tomás Abril

0. Explicação geral

O programa começa criando três variáveis que contem contextos: ContextPing, ContextPong e ContextMain.
Na função main cria-se um stack para o contexto ping e outro para o contexto pong, associam-se variáveis a cada contexto.
Em seguida o contexto atual é salvo em ContextMain e troca-se para o ContextPing. Com isso a função BodyPing é executada e inicia uma contagem de 0 a 3 imprimindo no terminal o valor dessa contagem. A cada contagem o contexto também é trocado para Pong que executa a função BodyPong e tem dentro dele uma contagem também de 0 a 3. A cada contagem dentro de Pong uma mensagem é impressa na tela e o contexto é novamente trocado para Ping.
Isso significa que a execução de Ping e Pong é alternada e continua até que o contador acabe.

1. Explicar o objetivo e os parâmetros de cada uma das quatro funções acima.

- getcontext(&a) : salva o contexto atual na variável a.
  No nosso programa é utilizado para inicializar os contextos.

- setcontext(&a) : restaura um contexto salvo anteriormente na variável a.
  Essa variavel tem que conter um contexto, no nosso programa não foi utilizado.

- swapcontext(&a,&b) : salva o contexto atual em a e restaura o contexto salvo anteriormente em b.
  É essa função que utilizamos para alternar entre o Ping e o Pong

- makecontext(&a, ...) : ajusta alguns valores internos do contexto salvo em a.
  Modifica o contexto: criando pilha, setando variáveis, definindo a função a ser chamada. Com essa função definimos o que são os contextos Ping e Pong antes de executá-los.

2. Explicar o significado dos campos da estrutura ucontext_t que foram utilizados no código.

- uc_stack.ss_sp contém a pilha alocada dinamicamente anteriormente.
- uc_stack.ss_size o tamanho de sua área na memoria.
- uc_stack.ss_flags uma flag de status de sinalização.
- uc_link seria um ponteiro para o contexto a ser executado depois que esse terminar, mas no nosso código não é utilizado.

3. Explicar cada linha do código de pingpong.c que chame uma dessas funções ou que manipule estruturas do tipo ucontext_t .

- ucontext_t ContextPing, ContextPong, ContextMain;
  Criando variáveis globais para guardar os contextos.

- swapcontext (&ContextPing, &ContextPong);
  Troca o contexto sendo executado de Ping para Pong dentro do loop.

- swapcontext (&ContextPing, &ContextMain) ;
  Quando o Ping é finalizado essa função volta a executar a main com a troca de contexto.

- swapcontext (&ContextPong, &ContextPing);
  Troca o contexto sendo executado de Pong para Ping dentro do loop.

- swapcontext (&ContextPong, &ContextMain) ;
  Quando o Pong é finalizado essa função volta a executar a main com a troca de contexto.

- getcontext (&ContextPing);
  Define o contexto atual como ContextPing.

- ContextPing.uc_stack.ss_sp = stack ;
  Associa o stack ao contexto.

- ContextPing.uc_stack.ss_size = STACKSIZE;
  Seta o tamanho do stack do contexto.

- ContextPing.uc_stack.ss_flags = 0;
  Flag de contexto.

- ContextPing.uc_link = 0;
  Link para o próximo contexto, no caso não tem nenhum.

- makecontext (&ContextPing, (void*)(*BodyPing), 1, "   Ping");
  Associa-se a função BodyPing ao contexto.

- getcontext (&ContextPong);
  Seta o contexto atual como contextPong.

- ContextPong.uc_stack.ss_sp = stack ;
  Associa o stack ao contexto.

- ContextPong.uc_stack.ss_size = STACKSIZE;
  Informa o tamanho do stack do contexto.

- ContextPong.uc_stack.ss_flags = 0;
  Flag de contexto.

- ContextPong.uc_link = 0;
  Link para o próximo contexto, no caso não tem nenhum.

- makecontext (&ContextPong, (void*)(*BodyPong), 1, " Pong");
  Associa-se a função BodyPong ao contexto.

- swapcontext (&ContextMain, &ContextPing);
  Inicia a execução do Ping, comecando o loop com a contagem. Ele irá alternar entre ping e pong até terminar.

- swapcontext (&ContextMain, &ContextPong);
  Como a contagem do Ping acaba antes da do Pong é necessário voltar mais uma vez para Pong para que ele finalize o seu loop.

4. Desenhar o diagrama de tempo da execução.

Contexto													
MainContext	x										x		x
PingContext		x		x		x		x		x			
PongContext			x		x		x		x			x	