

INE 5410 - Laboratório

AULA 01 – UTILIZANDO PASCAL-FC

1. INTRODUÇÃO

A linguagem Pascal-FC foi desenvolvida para dar suporte ao ensino de programação concorrente, incluindo mecanismos de controle da concorrência interessantes para o bom entendimento dos conceitos apresentados em sala de aula. Ela permite executar um conjunto de processos num ambiente de concorrência, de modo que estudantes entendam as dificuldades e os problemas associados à este tipo de programação.

O fato de ser baseado na linguagem Pascal, permite a prototipagem rápida de problemas e soluções utilizando mecanismos clássicos como os mencionados anteriormente.

2. OBJETIVO DA AULA

O objetivo desta aula prática é permitir ao estudante um primeiro contato com o ambiente de execução da linguagem Pascal-FC e experimentar problemas relacionados à execução de processos concorrentes nesta linguagem.

3. INSTALANDO A LINGUAGEM PASCAL-FC EM AMBIENTE WINDOWS

Apesar de existirem versões da linguagem Pascal-FC para outras plataformas, iremos instalar o ambiente para versões do Windows.

3.1. Instalando o Pascal-FC no Windows XP e anteriores

- Realizar o download do pacote Pascal-FC para o ambiente Windows no site da disciplina (link “Material”) e instalar o executável “editor-pascalfc-108.exe” do ambiente.

3.2. Instalando o Pascal-FC no Windows Vista e Seven

- Realizar o download do pacote Pascal-FC para o ambiente Windows no site da disciplina (link “Material”) e instalar o executável “editor-pascalfc-108.exe” do ambiente;
- Navegar até a pasta onde foi instalado o Pascal-FC (geralmente “C:\Arquivos de Programas\Editor PFC”) e localizar o arquivo “PFCEditor.exe”;
- Clicar com o botão direito do mouse no arquivo e escolher a opção “Propriedades”;
- Na janela de “Propriedades”, clicar sobre a aba “Compatibilidade” e ajustar a opção “Executar este programa como Administrador”;

- Finalmente, copie os arquivos “PFCCOMP.EXE” e “PINT.EXE” presentes na pasta onde os arquivos de instalação do Pascal-FC foram descompactados e copie para a pasta onde foi instalado o Editor Pascal-FC (geralmente, “C:\Arquivos de Programas\Editor PFC”) sobrescrevendo os arquivos de mesmo nome.

3.3. Testando a instalação

- Ative o Editor Pascal-FC e digite o seguinte código para testar a instalação:

```
PROGRAM teste;  
VAR n1,n2 : integer;  
BEGIN  
  WRITE('Entre com o primeiro número: ');  
  READLN(n1);  
  WRITE('Entre com o segundo número: ');  
  READLN(n2);  
  IF (n1 >= n2)  
    THEN WRITELN('O maior número é o ',n1)  
    ELSE WRITELN('O maior número é o ',n2);  
END.
```

- Executar o programa clicando sobre o menu “Programa”, opção “Ejecución justa” ou clicando diretamente sobre o ícone que representa uma balança equilibrada. Entre com os dados solicitados em formato inteiro.

4. OPERANDO COM PROCESSOS

Sendo uma linguagem orientada à Programação Concorrente, Pascal-FC permite realizar programas utilizando o conceito de processo, como um componente de programa que executa em ambiente concorrente com seus pares.

Edite as seguintes linhas de código no ambiente no editor do Pascal-FC:

```
PROGRAM Concorre;  
  
PROCESS Type TP(n : integer);  
BEGIN  
  WHILE (TRUE) DO  
    BEGIN  
      sleep(random(10));  
      writeln('Processo ',n,' executando!!!');  
    END;  
END;  
  
VAR  
  Proc : ARRAY[1..2] OF TP;  
  I : integer;  
BEGIN  
  COBEGIN  
    FOR I := 1 TO 2  
      DO Proc[I](I);  
    COEND;  
END.
```

Algumas explicações sobre o programa são necessárias:

- A palavra reservada `PROCESS` permite declarar um tipo de processo cujo código de execução é delimitado pelo bloco `BEGIN ... END`;
- `WHILE (TRUE)` determina uma malha infinita, sendo que o código delimitado pelo segundo bloco `BEGIN ... END` logo após o comando `WHILE` será repetido por um número indeterminado de vezes;
- `SLEEP` é uma função que, como o nome indica, provoca uma pausa na execução do processo, cuja duração tem valor aleatório entre 1 e 10 intervalos de tempo (definido pela execução da função `RANDOM(10)`);
- Durante sua execução, o processo apenas imprime uma notificação (imprimindo seu número) de que está em execução;
- Para a definição de diversos processos, foi declarada uma variável (ali denominada `Proc`) que é do tipo `ARRAY` (estrutura) de Processos, definindo apenas dois processos (`Proc[1]` e `Proc[2]`);
- O bloco `COBEGIN ... COEND` determina um bloco de execução em que processos são criados executarão concorrentemente... a malha `FOR` delimitada por este bloco é responsável pela criação e ativação da execução dos dois processos;
- Neste programa, como foram definidos apenas dois processos, espera-se que estes dois se alternem em suas execuções... isto poderá ser observado durante a execução do exemplo.

4.1. Executando o programa

- Execute o programa tal qual ele foi editado e observe o resultado da execução;
- Agora, aumente o número de processos do programa e execute para observar o que ocorre com um maior número de processos (aumente para nove, modificando o valor dos limites inferiores de 2 para 9 na declaração da variável `Proc` e na declaração da malha `FOR`).

5. EXERCÍCIOS

5.1. Exclusão Mútua

Caso seja necessário, modifique o programa da seção 4 para que o processo de impressão de cada processo não seja interrompido pela impressão de outro (exclusão mútua na impressão). Sugestão: faça uso de uma variável global de tipo booleano para garantir isso.

5.2. Sincronização de Processos

Tente modificar o programa da seção 4 para garantir a execução sequencial dos processos. Uma sugestão é utilizar uma variável global a ser acessada por todos os processos da seguinte maneira:

- Esta variável é inicializada a 1 antes do bloco `COBEGIN ... COEND` no programa;
- Cada processo, antes de entrar no processo de impressão (“execução”) testa o valor da variável para verificar se o seu valor é igual ao seu identificador (número do processo representado pela variável `n`);
- Caso a variável seja diferente do número do processo, este pula a execução (comando de impressão) e vai tentar novamente;

- Cada processo que conseguiu executar (imprimiu sua mensagem) realiza um incremento no valor desta variável de modo que o próximo processo possa então executar (caso o identificador do processo seja igual a 9, ao invés de incrementar o valor da variável, este reverte seu valor para 1).

5.3. Rede de Computadores

A partir do programa utilizado, escreva um novo programa que represente um cenário onde cada processo corresponde a um computador numa rede que utilize um barramento para encaminhar uma mensagem por uma rede... considere 9 processos (ou 9 computadores compondo esta rede)...

Represente a ação de enviar uma mensagem por uma instrução de impressão (WRITELN) de uma mensagem que indique o número do processo que está enviando.