

Apresentação

Tópicos Abordados no Treinamento

- Apresentação do Sistema
 - IHM – Interface Humano-Máquina
 - Principais Funcionalidades
- Ferramentas de Desenvolvimento:

<ul style="list-style-type: none"> • Base de dados de Strings <ul style="list-style-type: none"> # EditBdStrph # LoadBdStr • Configuração de tarefas <ul style="list-style-type: none"> # InstalaCtrl • Atualização arquivo DescrArqVar <ul style="list-style-type: none"> # IniArqVar • Configuração da base de dados <ul style="list-style-type: none"> # InstalaDemo 	<ul style="list-style-type: none"> • Configuração de eventos <ul style="list-style-type: none"> # DescrEvph # IniDescrEv • Configuração Descritor de Relatórios <ul style="list-style-type: none"> # DescrRelph • Configuração da IHM <ul style="list-style-type: none"> # ConflHMph • Seleção da Base de Simulação <ul style="list-style-type: none"> # selred
--	--

Copyright © 2012 Accenture. All rights reserved.

4

Apresentação

Tópicos Abordados no Treinamento

- Estrutura de Dados:
 - Arquitetura
 - Tipos de Variáveis (parâmetros, variáveis de processo, diárias e turno)
 - Declaração de Eventos
- Configuração:
 - Parâmetros
 - Eventos
 - Descritor Relatórios
 - Chamada na IHM
- Programação em C:
 - Tarefas (Mcp e Mcc)
 - Programas de operação
 - Relatórios (ciclo de controle, diários e turno)

Copyright © 2012 Accenture. All rights reserved.

5

Apresentação

Tópicos Abordados no Treinamento

- Preparação e Instalação do Upgrade:
 - Backup dos arquivos que serão alterados
 - Geração do arquivo de upgrade
 - Instalação do upgrade
- Projeto Final
 - Desenvolvimento de um algoritmo:
 - Declaração de Variáveis
 - Tarefa Mcp
 - Tarefa Mcc
 - Programa de Operação
 - Relatório de Ciclo de Controle
 - Relatório Diário e Turno
 - Instalação do algoritmo nos nodos de produção

Copyright © 2012 Accenture. All rights reserved.

6

Apresentação

Laboratório

- Partir o sistema de controle na máquina virtual QNX4_VM-1 e a interface de supervisão na QNX4_VM-3
- Apresentar os principais módulos e telas da IHM do Score
 - Pré-seleção
 - Configuração
 - Operação
 - Relatórios
 - Parâmetros de Linha e Cuba
 - Instantâneos
 - Ciclo de Controle e Históricos Diário/Turno
 - Eventos e Mensagens
 - Gráficos
 - Alarmes
 - Módulo Histórico

Copyright © 2012 Accenture. All rights reserved.

7

Apresentação

Laboratório (continuação ...)

- Apresentar alguns conceitos básicos da IHM
 - Base de dados de strings
 - Parâmetros de cuba e linha
 - Variáveis de processo
 - Eventos
 - Alarmes
 - Gráficos
- Preparar a máquina virtual para o desenvolvimento do sistema:
 - Instalar o runtime do Score (original CBA)
 - Instalar o compilador Watcom C e toolkit de desenvolvimento TCP/IP e photon
 - Instalar o arquivo **qmake.tar.F**
 - Instalar o toolkit de desenvolvimento **score_kit_des.tgz**
 - Instalar a base de dados de demonstração **cbabddemo.tgz**

Copyright © 2012 Accenture. All rights reserved.

8

Índice

Apresentação

Ferramentas de Desenvolvimento

Arquitetura do Sistema

Estruturas de Dados

Algoritmo de Controle

Criação de Opção de Operação

Criação de Relatórios

Preparação e Instalação de Upgrades

Projeto Final

Copyright © 2012 Accenture. All rights reserved.

9

Ferramentas de Configuração

EditBdStr - Edição da Base de Dados de Strings

1º Passo : carregar o photon

ph

2º Passo : chamar o editor pela [Barra de Ferramentas](#) selecionando [Grupo Score Tools](#) → [EditBdStr](#), ou via shell pelo comando abaixo:

/score/util/EditBdStr



Copyright © 2012 Accenture. All rights reserved.

10

Ferramentas de Configuração

EditBdStr - Edição da Base de Dados de Strings

- A tela do programa é carregada em branco. Na régua do rodapé há botões de paginação, manipulação de informações e saída:



- Botões Paginação:** PGUP, PGDN, HOME, END
- Botões Manipulação:** tratamento de informações
 - READ** : carregar a base de dados selecionada para memória do editor
 - WRITE** : gravar os dados da memória para arquivo em disco
 - GOTO** : posicionar na linha especificada
 - SEARCH** : pesquisar o padrão de string especificado
 - DIFF** : comparar as informações de memória com o arquivo selecionado
 - COPY** : copiar informações do arquivo selecionado para a base de dados aberta em memória
 - LOAD** : não utilizada
 - QUIT** : sair do programa

Copyright © 2012 Accenture. All rights reserved.

11

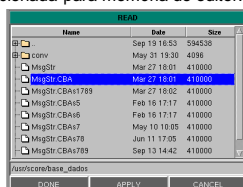
Ferramentas de Configuração

EditBdStr - Edição da Base de Dados de Strings

- READ** : carregar a base de dados selecionada para memória do editor:

Arquivos de Bases de Dados

- Diretório:** /score/base_dados
- Bases de Dados:**
 - MsgStr**: usado pelo Score
 - MsgStr.CBA**: reduções II, III e IV
 - MsgStr.CBAs1789**: reduções I, V VI e VII



- Características da Tela:**
 - Tree-View**: área para seleção do arquivo de base de dados
 - Path**: área para seleção da base de dados para configurar
 - Botões:** **DONE** – Finalizar, **APPLY** – Confirmar, e **CANCEL** – Sair ou Abandonar

Copyright © 2012 Accenture. All rights reserved.

12

Ferramentas de Configuração

EditBdStr - Edição da Base de Dados de Strings

- WRITE** : salvar a base de dados da memória no disco. A operação é confirmada pela mensagem ao lado



- GOTO** : posicionar na linha definida pela tela ao lado para alteração da string.



- SEARCH** : pesquisar a string informada pela tela ao lado



Copyright © 2012 Accenture. All rights reserved.

13

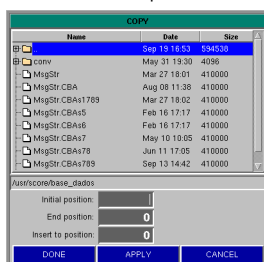
Ferramentas de Configuração

EditBdStr - Edição da Base de Dados de Strings

- COPY** : copiar uma área da base de dados selecionada pela tela abaixo para memória do editor

- Características da Tela:

- Tree-View** : área de seleção do arquivo de base de dados
- Path** : área para definição do path do arquivo
- Coordenadas da Transferência**:
 - Inicial position**: posição inicial da área copiada
 - End position**: posição final da área copiada
 - Insert to position**: posição inicial que a área será inserida



- Botões**: **DONE** – Finalizar, **APPLY** – Confirmar e **CANCEL** – Sair ou Abandonar

Copyright © 2012 Accenture. All rights reserved.

14

Ferramentas de Configuração

EditBdStr - Edição da Base de Dados de Strings



- LOAD** : essa funcionalidade é implementada pelo programa `/score/util/LoadBdStr` executado através do shell do QNX, não está implementada nessa interface
Esse botão funciona somente quando o programa **EditBdStr** é ativado a partir do diretório `/score/util`
- QUIT** : sair do editor

Copyright © 2012 Accenture. All rights reserved.

15

Ferramentas de Configuração

EditBdStr - Edição da Base de Dados de Strings

[usr\score\base_dados\MsgStr.CBAs1789]		
4047	Ativa/Desativa Tabela de Efeito Anódico	40
4048	Ativa/Desativa Tabela de Partida	32
4049		80
4050		80

Edição de Strings

- Uma mensagem é formada por 2 atributos da base de dados de strings:
 - Posição:** posição sequencial da mensagem na base de dados
 - Mensagem:** conteúdo da string que é exibido na tela ou evento
 - Tamanho:** tamanho da mensagem, **default = 80**
- A string sempre deve ser salva após a alteração

Copyright © 2012 Accenture. All rights reserved.

16

Ferramentas de Configuração

Laboratório

- Incluir a chamada da ferramenta EditBdStr na régua de ferramentas do photon:
 - Criar o grupo Score Tools
 - Incluir a chamada do EditBdStrph
- Carregar o editor de strings
 - Carregar a base de strings da redução que está em execução
 - Posicionar na string **620**
 - Incluir a palavra **Novo** nas strings **623** e **627**
 - Salvar as alterações
- Retirar a IHM de supervisão e partir novamente
 - Verificar se a alteração que você fez apareceu na tela de relatórios

Copyright © 2012 Accenture. All rights reserved.

17

Ferramentas de Configuração

LoadBdStr - Instalação da Base de Dados de Strings

1º Passo : Copiar a base de dados de string **PathStr** alterada para o diretório **/score/base_dados** do micro de controle da redução em execução, onde **PathStr** é :

- /score/base_dados/MsqStr.CBA:** reduções II, III e IV
- /score/base_dados/MsqStr.CBAs1789:** reduções I, V, VI e VII

2º Passo : carregar a base de dados de string atualizada pelo utilitário **LoadBdStr** passando como parâmetro o arquivo **PathStr**, conforme a sintaxe abaixo:

```
# /score/util/LoadBdStr /score/base_dados/PathStr
```

O utilitário **LoadBdStr** copia a base de dados **PathStr** para o arquivo lido pelo sistema **/score/base_dados/MsgStr** e carrega esse arquivo atualizado na memória principal se o sistema estiver em execução.

Copyright © 2012 Accenture. All rights reserved.

18

Ferramentas de Configuração

Laboratório

- Carregar a respectiva base de dados de strings do sistema que está em execução no na QNX4_VM-1 (controle)
 - Copiar a base de dados de strings alterada no laboratório anterior para o diretório `/score/base_dados` do nodo de controle
 - Carregar a base de dados de strings pelo utilitário **LoadBdStr**:

```
# /score/util/LoadBdStr /score/base_dados/PathStr
```
- Retirar a IHM de supervisão e partir novamente
 - Verificar se a alteração que você fez apareceu na tela de relatórios

Copyright © 2012 Accenture. All rights reserved.

19

Ferramentas de Configuração

DescrEvph – Configuração do Descritor de Eventos

1º Passo : carregar o photon

```
# ph
```

2º Passo : chamar o editor pela **Barra de Ferramentas** selecionando **Grupo Score Tools** → **DescrEvPh**, ou via shell pelo comando abaixo:

```
# /score/util/DescrEvph
```

Esse utilitário edita o arquivo `/score/base_dados/descr_ev`



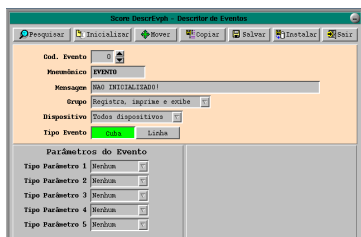
Copyright © 2012 Accenture. All rights reserved.

20

Ferramentas de Configuração

DescrEvph – Configuração do Descritor de Eventos

- A tela do programa é carregada, a qual é formada por 3 áreas distintas:
 - Régua de Botões
 - Configuração do evento
 - Configuração tipo dos 5 parâmetros do evento



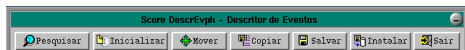
Copyright © 2012 Accenture. All rights reserved.

21

Ferramentas de Configuração

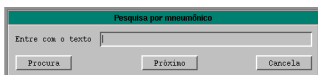
DescrEvph – Configuração do Descritor de Eventos

Régua de Botões

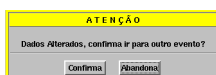


- **Pesquisar** : localizar um evento pelo mneumônico ou mensagem definida na tela abaixo:

- **Procura** : localiza a primeira ocorrência do texto
- **Próximo** : localiza próxima ocorrência do texto
- **Cancela** : sai da tela de pesquisa e encerra



- **Inicializar** : inicializar todos registros de evento com os valores default. Quando essa inicialização é feita, a tela ao lado é exibida toda vez que muda a posição do registro de evento



Copyright © 2012 Accenture. All rights reserved.

22

Ferramentas de Configuração

DescrEvph – Configuração do Descritor de Eventos

- **Mover** : move o registro do evento corrente para a posição informada na tela ao lado



- **Copiar** : copia o registro do evento corrente para a posição informada na tela ao lado



- **Salvar** : grava os dados de memória no arquivo `/score/base_dados/descr_ev`
- **Instalar** : grava os dados de memória no arquivo `/score/base_dados/descr_ev` em arquivo e memória do comum. Essa opção não é utilizada pois normalmente instala-se esse arquivo via upgrade nos micros através do utilitário `IniDescrEv`
- **Sair** : sair do programa

Copyright © 2012 Accenture. All rights reserved.

23

Ferramentas de Configuração

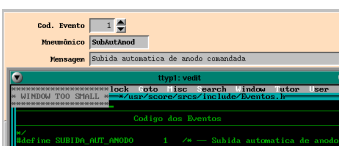
DescrEvph – Configuração do Descritor de Eventos

Configuração do Evento

- **Cod. Evento** : posição do evento no arquivo (0 a 200). Essa posição é o mesmo nº do define desse evento do prólogo Eventos.h

A faixa de reservada para eventos do cliente é de 65 a 200.

- **Mneumônico** : código do evento que aparece exibido no relatório de eventos
- **Mensagem** : mensagem do evento exibida na linha de alarmes
- **Grupo** : tratamento do evento de acordo com as opções *Registra Imprime e Exibe*, *Registra Exibe*, *Registra Imprime*, *Apenas Registra*, *Não Registra*



Copyright © 2012 Accenture. All rights reserved.

24

Ferramentas de Configuração

DescrEvph – Configuração do Descritor de Eventos

- **Dispositivo** : dispositivos de exibição da mensagem do evento, entre as opções *Todos Dispositivos*, *Exceto Graf Cuba*, *Exceto Ev Inst e Exceto Ev e Graf*
- **Tipo Evento** : indica se é evento de cuba ou linha

Tipos de Parâmetros do Evento

- **Tipo Parâmetro 1 a 5** : configuração do tipo do 1º ao 5º parâmetro do evento de acordo com as opções *Char*, *Int*, *Long*, *Float*, *Str* e *Nenhum* quando o parâmetro não for utilizado.

Copyright © 2012 Accenture. All rights reserved.

25

Ferramentas de Configuração

Laboratório

- Incluir a chamada da ferramenta EditBdStrph no grupo Score Tools da régua de ferramentas do photon
- Criar um evento com o mneumônico **EvTeste**:
 - Carregar o descritor de eventos **DescrEvph** e criar um evento com o mneumônico **EvTeste** na 1ª posição disponível do descritor de eventos **descr_ev**
 - Copiar o descritor de eventos alterado para para o diretório */score/base_dados* do nodo de controle
- Retirar a IHM de supervisão e partir novamente
 - Verificar se a alteração que você fez apareceu na tela de relatórios

Copyright © 2012 Accenture. All rights reserved.

26

Ferramentas de Configuração

IniDescrEv – Instalação do Descritor de Eventos

1º Passo : instalar o descritor de eventos atualizado no comum pelo utilitário **IniDescrEv** conforme a sintaxe abaixo:

```
# /score/uti/IniDescrEv -e -v
```

- O utilitário LoadBdStr copia os dados do arquivo */score/base_dados/dscr_ev* para sua respectiva estrutura de dados do arquivo */score/base_dados/comum* em disco e na memória principal se o sistema estiver executando

Copyright © 2012 Accenture. All rights reserved.

27

Ferramentas de Configuração

Laboratório

- Instalar o descritor de eventos alterado na base de dados do sistema que está em execução no na QNX4_VM-1 (controle)
 - Instalar o descr_ev já copiado para o diretório /score/base_dados do nodo de controle pelo utilitário **IniDescrEv**:
`# /score/util/IniDescrEv -e -v`
- Retirar a IHM de supervisão e partir novamente
 - Verificar se a alteração que você fez apareceu na tela de relatórios

Copyright © 2012 Accenture. All rights reserved.

28

Ferramentas de Configuração

DescrRelph – Configuração do Descritor de Relatórios

1º Passo : carregar o photon

```
# ph
```

2º Passo : chamar o editor pela **Barra de Ferramentas** selecionando **Grupo Score Tools → DescrRelPh**, ou via shell pelo comando abaixo:

```
# /score/util/DescrRelph
```



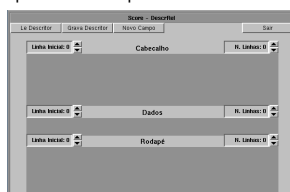
Copyright © 2012 Accenture. All rights reserved.

29

Ferramentas de Configuração

DescrRelph – Configuração do Descritor de Relatórios

- A tela do programa é carregada, a qual é formada por 4 áreas distintas:
 - Régua de Botões
 - Configuração do cabeçalho
 - Configuração da área de dados
 - Configuração do rodapé
- A configuração do tamanho das áreas:
 - **Linha Inicial**: 1ª linha da área
 - **N. Linhas**: número de linhas da área configurada
- **Observações**:
 - **Limite Cabeçalho e Rodapé** : 5 linhas
 - **Limite do Relatório**: 24 linhas
 - $\text{LinhaInicial(dados)} = \text{LinhaInicial(Cabeçalho)} + \text{N.Linhas(Cabeçalho)}$
 - $\text{LinhaInicial(Rodapé)} = \text{LinhaInicial(Dados)} + \text{N.Linhas(Dados)}$



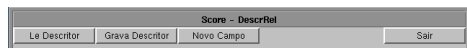
Copyright © 2012 Accenture. All rights reserved.

30

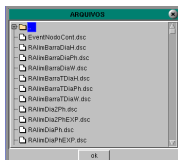
Ferramentas de Configuração

DescrRelph – Configuração do Descritor de Relatórios

Régua de Botões



- **Le Descritor** : selecionar o descritor que será carregado. Quando seleciona essa opção, a tela ao lado é exibida com a lista de arquivos do diretório `/score/descr`.



• Observações:

- O nome do descritor sem a extensão `.desc` deve ter o nome do respectivo programa executável.
- Geralmente utilizamos como template um descritor semelhante ao relatório que criamos para agilizar o processo um novo descritor nem salvar com outro nome

Copyright © 2012 Accenture All rights reserved.

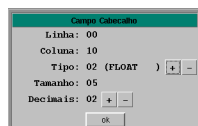
31

Ferramentas de Configuração

DescrRelph – Configuração do Descritor de Relatórios

- **Novo Campo** : cria um novo campo no descritor do relatório. O campo é criado através dos seguintes passos:

1. Preencher com **x** o espaço do novo campo, como na tela ao lado
2. Marcar o campo com o mouse, conforme a tela ao lado
3. Selecionar o botão **Novo Campo** para configurar o campo
4. Configurar o Tipo do campo as opções **INTEIRO** (int), **Float**, **_ASCII** (char), **LONGO** (long), **UNASIGNED**, **DATA**, **HORA**.



Copyright © 2012 Accenture All rights reserved.

32

Ferramentas de Configuração

DescrRelph – Configuração do Descritor de Relatórios

- **Grava Descritor** : salvar o descritor em disco. Quando é um descritor novo, a tela ao lado é exibida para receber o nome na 1ª vez que o arquivo é salvo.



- **Sair** : sair do programa

Copyright © 2012 Accenture All rights reserved.

33

Ferramentas de Configuração

Laboratório

- Incluir a chamada da ferramenta DescrRelph no grupo Score Tools da régua de ferramentas do photon
- Criar um arquivo descritor para o relatório de ciclo de controle do toolkit de desenvolvimento.

Copyright © 2012 Accenture. All rights reserved.

34

Ferramentas de Configuração

ConflHMph – Arquitetura da Tela da IHM

1	6	11	16
2	7	12	7
3	8	13	18
4	9	14	19
5	10	15	20

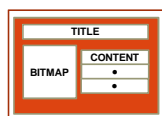
- **20 Botões:**
 - 5 x 4 (Linhas x Colunas)
 - Configurados via **ConflHMph**
- **Tipos de Tela:**
 - **RelTela (5)** : relatórios diversos
 - **RelGraf (2)** : relatórios gráficos
 - **Operacao (5)**: opções de operação
 - **Configura (2)**: opções de configuração

Copyright © 2012 Accenture. All rights reserved.

35

Ferramentas de Configuração

ConflHMph – Arquitetura da Tela da IHM



- **Atributos Configurados via ConflHMph**
 - **Bitmap**: ícone do botão
 - **Title**: título do botão (tipo de tela)
 - **Content**: descrição da opção executada

- **Relação entre ConflHMph x EditBdStrph**
 - **Title e Content:**
 - Configurados na base de dados de strings **MsgStr**
 - **ConflHMph**: configura na IHM a posição da respectiva mensagem cadastrada na IHM
 - Espaço reservado sequencialmente para botões no **MsgStr**
 - **Botão**: 4 linhas de mensagens (1 Title + 3 Content)
 - **Tela** : 20 botões → 4 x 20 = 80 linhas de mensagens

Copyright © 2012 Accenture. All rights reserved.

36

Ferramentas de Configuração

ConfIHMph – Cálculo da Posição dos Textos do Botão

Posição Inicial P_t	
Tela	P_t
RelTela1	620
RelTela2	700
RelTela3	780
RelTela4	860
RelTela5	940
RelGraf1	1020
RelGraf2	1100
Operacao1	1180
Operacao2	1260
Operacao3	1340
Operacao4	1420
Operacao5	1500
Configura1	1580
Configura2	1660

• Cálculo da Posição Inicial do Botão:

Formula: $P_b = P_t + ((O_b - 1) \times 4) \rightarrow O_b = \{1, 2, 3, \dots, 20\}$

onde: $P_b \rightarrow$ posição inicial do botão (Title)

$P_t \rightarrow$ posição inicial da tela (tabela ao lado)

$O_b \rightarrow$ ordem (posição) do botão na tela

• Exemplo: Tela **Operacao1**, $P_t = 1180$

• 1º Botão: $O_b = 1 \rightarrow P_b = 1180 + (1 - 1) \times 4 = 1180$

• 2º Botão: $O_b = 2 \rightarrow P_b = 1180 + (2 - 1) \times 4 = 1184$

• 20º Botão: $O_b = 20 \rightarrow P_b = 1180 + (20 - 1) \times 4 = 1256$

Copyright © 2012 Accenture. All rights reserved.

37

Ferramentas de Configuração

Laboratório

- Abrir a IHM de supervisão do sistema e selecionar a tela de operação
- Abrir a base de dados de strings **MsgStr** pelo **EditBdStr** e localizar os atributos **Title** e **Content** do 1º botão de operação

Copyright © 2012 Accenture. All rights reserved.

38

Ferramentas de Configuração

ConfIHMph – Configuração da IHM

1º Passo : carregar o photon

ph

2º Passo : chamar o editor pela **Barra de Ferramentas** selecionando **Grupo Score Tools** → **ConfIHMph**, ou via shell pelo comando abaixo:

/score/util/ConfIHMph



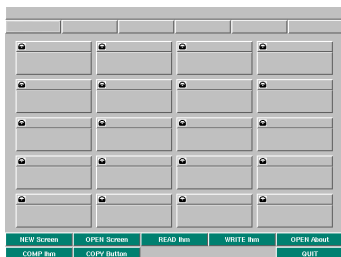
Copyright © 2012 Accenture. All rights reserved.

39

Ferramentas de Configuração

ConfIHMph – Configuração da IHM

- A tela do programa é carregada, a qual é formada por 2 áreas distintas:
 - Área da tela carregada
 - Réguas de Botões



Copyright © 2012 Accenture. All rights reserved.

40

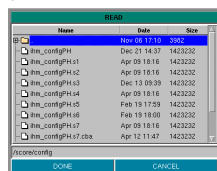
Ferramentas de Configuração

ConfIHMph – Configuração da IHM

Réguas de Botões

NEW Screen	OPEN Screen	READ Btn	WRITE Btn	OPEN /about
COMP Btn	COPY Button			QUIT

- READ Screen** : selecionar a IHM do diretório `/score/config` pela tela abaixo



IHM x Sala

IHM de Supervisão	
Path	Sala
/score/config/ihm_configPH.s1	IHM da sala 125 kA I
/score/config/ihm_configPH.s2	IHM da sala 125 kA II
/score/config/ihm_configPH.s3	IHM da sala 125 kA III
/score/config/ihm_configPH.s4	IHM da sala 125 kA IV
/score/config/ihm_configPH.s7	IHM da sala 125 kA V
/score/config/ihm_configPH.s8	IHM da sala 125 kA VI
/score/config/ihm_configPH.s9	IHM da sala 125 kA VII

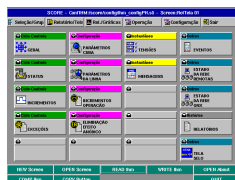
Copyright © 2012 Accenture. All rights reserved.

41

Ferramentas de Configuração

ConfIHMph – Configuração da IHM

- A IHM selecionada é carregada conforme a tela ao lado
- OPEN Screen** : selecionar a tela da IHM que será carregada a partir do combo Box exibido com as seguintes opções:
 - Rel Tela 1
 - Rel Tela 2
 - RelGráfico
 - Operação
 - Configuração
- NEW Screen** : criar uma nova tela de IHM. Quando essa opção é selecionada um novo gabarito da tela todo em branco é exibido para incluir os botões.
- WRITE Ihm** : salvar a IHM carregada em disco. Quando essa operação é executada, a mensagem ao lado é exibida informando sua conclusão.



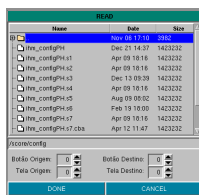
Copyright © 2012 Accenture. All rights reserved.

42

Ferramentas de Configuração

ConflHMph – Configuração da IHM

- **COPY Button** : copia o botão da IHM selecionada (origem) pela tela ao lado para a IHM aberta (destino) de acordo com as coordenadas abaixo:
- **Botão Origem**: botão que será copiado domínio de 0 a 19.
- **Tela Origem**: tela de onde o botão será copiado
- **Botão Destino**: posição da tela para onde o botão será copiado
- **Tela Destino**: tela para onde o botão será copiado



Copyright © 2012 Accenture. All rights reserved.

43

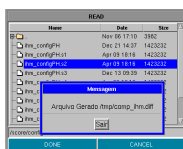
Ferramentas de Configuração

ConflHMph – Configuração da IHM

- A identificação das telas de IHM variam de uma sala para a outra. Na tabela ao lado podemos ver um exemplo a lista de telas da IHM da sala I

Telas da IHM			
Nº	Tela	Nº	Tela
0	RefTela 01	5	RefTela 03
1	RefTela 02	6	ManutScore
2	RefGrafico	7	Operacao2
3	Operacao	8	OperacaoMovel
4	Configuracao	9	RefTela04

- **COMP Ihm** : comparar a IHM em memória com a IHM selecionada na tela ao lado. O resultado da comparação é registrado em arquivo cujo *path* é informado pela mensagem ao lado.



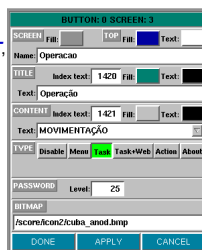
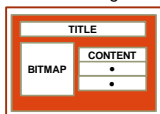
Copyright © 2012 Accenture. All rights reserved.

44

Ferramentas de Configuração

ConflHMph – Configuração da Chamada do Botão

- **1º Passo** : selecionar a IHM desejada e carregar na memória através do botão **READ Ihm**.
- **2º Passo** : selecionar a tela desejada e carregar na memória através do botão **OPEN Screen**.
- **3º Passo** : selecionar o botão e configurar seu lay-out pelos campos **TITLE**, **CONTENT**, e **BITMAP** conforme gabarito abaixo.



Copyright © 2012 Accenture. All rights reserved.

Ferramentas de Configuração

ConfIHMph – Configuração da Chamada do Botão

- **4º Passo** : definir nível de senha de acesso pelo campo **PASSWORD** de 0 a 255. Se o **Level** é maior que 0, a exibição do ícone de cadeado é habilitada. Geralmente essa configuração segue o seguinte padrão:

- **Nível 25**: programas de operação
- **Nível 50**: módulo histórico
- **Nível 75**: módulo de configuração, exceto cadastro de usuários
- **Nível 255**: Hard-Copy, cadastro de usuários e sair do Score

- **5º Passo** : definir o tipo de ação do botão entre as opções **Disable**, **Menu**, **Task** e **Action**.

- **Disable**: desabilita o botão

- **Menu**: exibe o menu selecionado no combo-box abaixo da régua de botões

- **Task**: configura a criação do programa



Copyright © 2012 Accenture. All rights reserved.

46

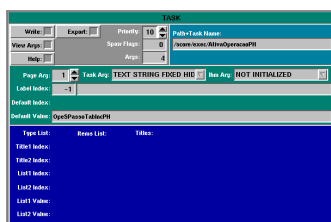
Ferramentas de Configuração

ConfIHMph – Configuração da Chamada do Botão

- **Action**: executa a ação selecionada no combo-box entre as opções:

- **Pots Selection**: exibe cubas selecionadas antes da ação
- **End System**: sai da IHM de supervisão e encerra sua execução

- **6º Passo** : configurar os parâmetros de criação do programa quando for tipo **Task**.



Copyright © 2012 Accenture. All rights reserved.

47

Ferramentas de Configuração

ConfIHMph – Configuração da Ativação do Programa

- A ativação dos programas configurada a através de 3 áreas da tela:

- **Interface**: flags de interface



- **Criação Programa**: flags e **path** de criação



- **Argumentos**: parâmetros passados como argumentos na criação do programa



Copyright © 2012 Accenture. All rights reserved.

48

Ferramentas de Configuração

ConflHMph – Configuração da Ativação - Interface

- **Interface**: a configuração de chamada da interface é feita através dos seguintes flags:
 - **Write**: grava os parâmetros como default para próxima chamada
 - **View Args**: exibe os argumentos antes de chamar o programa
 - **Export**: pede confirmação antes de ativar o programa
 - **Help**: habilita arquivo de ajuda do programa *NomeProg*, cujo path será */score/help/NomeProg.hlp*



Copyright © 2012 Accenture. All rights reserved.

49

Ferramentas de Configuração

ConflHMph – Configuração da Ativação - Criação

- **Criação Programa**: a configuração dos parâmetros de criação é feita pelas seguintes opções:

- **Nome + Task Name**: path do programa que será criado.

Observação:

O path dos programas de operação é configurado conforme a tela abaixo com o path */score/exec/AtivaOperacaoPH* e o nome do programa *NomeProg* sem path no campo **Default Value**.



No exemplo de tela vemos a criação do programa *OpeSPassoTabIncPH*

Copyright © 2012 Accenture. All rights reserved.

50

Ferramentas de Configuração

ConflHMph – Configuração da Ativação - Criação

- **Prioridade**: prioridade de execução do programa, geralmente executam com prioridade 10
- **Spawn Flags**: flags de criação da tarefa passados no comando *spawn*
- **Num Args**: número de argumentos relacionados à criação do programa que foram configurados na IHM. Importante destacar que esse parâmetro não tem nenhuma relação com a variável *argc* utilizada no programa em C.

Copyright © 2012 Accenture. All rights reserved.

51

Ferramentas de Configuração

ConflHMph – Configuração da Ativação - Criação

- **Argumentos** : a configuração dos *n* argumentos definidos em **Args** é feita pelas seguintes opções:

- **Page Arg** : seleciona a pagina de configuração do argumento
- **Task Arg** : tipo do argumento passado para o programa

Arguments Type Defined	
TASK ARG	Description
ALGORITM	Algoritmo definido na tela de pré-seleções
DATE END	Data Final definida na tela de pré-seleções
DATE INITIAL	Data Inicial definida na tela de pré-seleções
DAY EXCLUDED	Lista de datas excluídas definida na tela de pré-seleções
LIST	Tela gráfica com botões de opções associado aos atributos de configuração de lista
OPERATOR	Identificação do operador passada durante o pedido de senha para o acesso à opção desejada.
POTS EXCLUDED	Lista de cubas excluídas definida na tela de pré-seleções

Copyright © 2012 Accenture All rights reserved.

52

Ferramentas de Configuração

ConflHMph – Configuração da Ativação - Criação

Continuação dos tipos de argumentos **Task Arg...**

Arguments Type Defined	
TASK ARG	Description
POTS KEYBOARD	Faixa de cubas definida via teclado acionado no rodapé da tela de chamada da opção.
POTS SELECTION	Faixa de cubas definida na tela de pré-seleções
POTS STRING	Faixa de cubas definida no formato de string.
POTS STRING	Faixa de cubas definida no formato de string.
REDUCTION	Número da redução do projeto
SHIFT	Turno definido na tela de pré-seleções
TEXT STRING	Tipo alfanumérico definido através de teclado alfabético que é vinculado aos campos Label Index e Default Index
TEXT STRING FIXED	Permite a passagem de um parâmetro interno fixo de execução do programa.
TEXT STRING FIXED HIDDEN	Permite a passagem de um parâmetro interno fixo de execução do programa oculto do usuário. Geralmente esse tipo é utilizado para passar argumentos para o programa que está sendo ativado.

Copyright © 2012 Accenture All rights reserved.

53

Ferramentas de Configuração

ConflHMph – Configuração da Ativação - Criação

- **IHM Arg** : tipo do argumento passado para a IHM conforme tabela abaixo

Arguments Type Defined	
IHM ARG	Description
CONFIRMS QUESTION	Exibe pergunta definida no campo Default Index para confirmar a execução, cuja resposta deve ser sempre Confirma ou Abandona .
TITLE	Título exibido no topo da tela de chamada da opção, que é definido no campo Default Index e não é passado para o programa acionado.
HELP	Determina se a opção terá arquivo de ajuda. Se essa opção estiver selecionada o arquivo de ajuda deverá ter o mesmo nome do módulo executável com a extensão “.hlp” , ficando no diretório /score/help

- **Label Index** : nome do campo da tela
- **Default Index e Default Value** : esses itens representam o valor do argumento, onde **Default Index** é a identificação do argumento exibido ao usuário na tela e **Default Value** o respectivo valor repassado ao programa como argumento e oculto ao usuário

Copyright © 2012 Accenture All rights reserved.

54

Ferramentas de Configuração

ConflHMph – Configuração da Ativação – Criação (Lista)

- **Task Arg: List** : o argumento tipo lista deve ter sempre **Type List : 0** para possibilitar a configuração dos demais parâmetros da lista, pois **Type List : 1** especifica a lista utilizada exclusivamente no relatório de eventos montada através do programa da IHM.

A lista é configurada através dos parâmetros da tela abaixo:

- **Items List** : nº de itens da lista que o campo vai receber (máximo 15)
- **Titles** : nº de áreas da tela (máximo 2), onde cada área equivale a 1 lista.
- **Title1 Index/Title2 Index** : título da 1ª e 2ª área respectivamente
- **List1 Value/List2 Value** : respectivos tags dos botões passados como **argv[]** para os programas de operação de botões da 1ª e 2ª área (separados por vírgula)

Copyright © 2012 Accenture. All rights reserved.

55

Ferramentas de Configuração

Resumo de Configuração da Chamada

- Resumo do procedimento de configuração da chamada:
 1. Selecionar a tela e o botão a configurar
 2. Configurar as condições de chamada da interface
 3. Configurar a chamada do programa
 4. Configurar os atributos dos argumentos de chamada

Copyright © 2012 Accenture. All rights reserved.

56

Ferramentas de Configuração

Laboratório

- Incluir a chamada da ferramenta ConflHMph no grupo Score Tools da régua de ferramentas do photon
- Incluir um botão de chamada do programa de operação de exemplo do toolkit de desenvolvimento

Copyright © 2012 Accenture. All rights reserved.

57

Ferramentas de Instalação

InstalaCtrl - Configuração de Tarefas Residentes

- Chamar o configurador pelo comando abaixo:

```
# /score/util/InstalaCtrl
```

```
STOR - SCORE 4.1 - INSTALACAO DE MODULOS DE CONTROLE
Sistema Score: DESATIVADO
Modulo MCC
M1 - /score/mcc/McScore 581 678 Prioridade: 14
M2 - /score/mcc/McScore 581 678 Prioridade: 14
M3 - /score/mcc/McScore 581 678 Prioridade: 14
M4 - /score/mcc/McScore 581 678 Prioridade: 14
M5 - /score/mcc/McScore 581 678 Prioridade: 14
Modulo MCP
M1 - /score/mcc/McScore 581 678 Prioridade: 14
M2 - /score/mcc/McScore 581 678 Prioridade: 14
M3 - /score/mcc/McScore 581 678 Prioridade: 14
M4 - /score/mcc/McScore 581 678 Prioridade: 14
M5 - /score/mcc/McScore 581 678 Prioridade: 14
ESC-Sair F8-Reiniciar F9=Ativar F10=Confirmar
```

- Tela composta por 3 áreas:
 - Módulos MCC:** área para configurar a partida de 6 tarefas MCC
 - Módulos MCP:** área para configurar a partida de 6 tarefas MCP
 - Rodapé:** régua de botões para alterar o cadastro e sair

Copyright © 2012 Accenture. All rights reserved.

58

Ferramentas de Instalação

InstalaCtrl - Configuração de Tarefas Residentes

- F8 = Reiniciar** : selecionar essa opção para reiniciar uma tarefa pela tela abaixo:

```
Modulo de controle (1=MCC, 2=MCP): 2
Numero do modulo: 2
ESC=Aborta F10=Confirma
```

- Configuração dos parâmetros de reinicialização:
 - Modulo de controle (1=MCC, 2=MCP):** seleciona **1** para programa de ciclo de controle (MCC) ou **2** para ciclo de leitura (MCP)
 - Numero do módulo:** seleciona de **1** a **6** para informar a posição do programa na base de dados
 - Confirmar pelo botão **F10=Confirma** ou abandonar por **ESC=Aborta**

Copyright © 2012 Accenture. All rights reserved.

59

Ferramentas de Instalação

InstalaCtrl - Configuração de Tarefas Residentes

- F9 = Ativar** : selecionar a opção para instalar ou ativar o programa pela tela abaixo:

```
Modulo de controle (1=MCC, 2=MCP): 2
Numero do modulo: 2
Nome do modulo (com path):
Prioridade: 14
Acao (1=Ativar e Instalar, 2=So Instalar): 1
ESC=Aborta F10=Confirma
```

- Configuração dos parâmetros de instalação:
 - Modulo de controle (1=MCC, 2=MCP):** seleciona **1** para programa de ciclo de controle (MCC) ou **2** para ciclo de leitura (MCP)
 - Numero do módulo:** seleciona de **1** a **6** para informar a posição do programa na base de dados
 - Nome do módulo (com path):** **path** completo do programa instalado
Esse path errado desativa o sistema imediatamente na partida
 - Prioridade:** **14** → MCP e **18** → MCC
 - Ação :** **1** → instalar na base de dados e ativar ou **2** → somente instalar
 - Confirmar pelo botão **F10=Confirma** ou abandonar por **ESC=Aborta**

Copyright © 2012 Accenture. All rights reserved.

60

Ferramentas de Instalação

InstalaCtrl - Configuração de Tarefas Residentes

- **F10 = Desativar** : selecionar essa opção para desinstalar ou desativar o programa pela tela abaixo:

```

Modulo de controle (1=MCC,2=MCP): 2
Numero do modulo: 2
Acao (1=Desativar e Desinstalar,2=So desativar): 1
ESC=Aborta      F10=Confirma

```

- Configuração dos parâmetros de desinstalação:
 - **Modulo de controle (1=MCC, 2=MCP)**: seleciona **1** para programa de ciclo de controle (MCC) ou **2** para ciclo de leitura (MCP)
 - **Numero do módulo**: seleciona de **1 a 6** para informar a posição do programa na base de dados
 - **Ação** : **1** → desativar e desinstalar da base de dados ou **2** → somente desativar
- Confirmar pelo botão **F10=Confirma** ou abandonar por **ESC=Aborta**
- **ESC=Sair** : selecionar essa opção para sair do configurador

ESC=Sair F8=Reiniciar F9=Ativar F10=Desativar_

Copyright © 2012 Accenture. All rights reserved.

61

Ferramentas de Instalação

Laboratório

- Alterar o path do programa **Mcp** e carregar a alteração na base de dados
- Verificar o que acontece na partida do sistema

Copyright © 2012 Accenture. All rights reserved.

62

Ferramentas de Instalação

selred - Seleção da Base de Dados de Simulação

- A plataforma de desenvolvimento pode ser executada para qualquer redução, cuja base de dados deve ser previamente selecionada.
- As bases de dados de simulação das reduções já vem configuradas com os drivers de demonstração para testar o sistema. Essas bases de dados foram organizadas nos diretórios do quadro abaixo.

Diretório de Base de Dados de Simulação	
Diretório	Conteúdo
/score/base_dados_r1	Base de dados de redução I
/score/base_dados_r2	Base de dados de redução II
/score/base_dados_r3	Base de dados de redução III
/score/base_dados_r4	Base de dados de redução IV
/score/base_dados_r7	Base de dados de redução V
/score/base_dados_r8	Base de dados de redução VI
/score/base_dados_r9	Base de dados de redução VIII

Copyright © 2012 Accenture. All rights reserved.

63

Ferramentas de Instalação

selred - Seleção da Base de Dados de Simulação

- A base de dados da redução que será testada é selecionada pelo comando **selred** passando como parâmetro o número da redução:
 - # selred Redução** ► deve ser executado com o sistema desativado
 - # selred 3** ► configura sistema para rodar redução III

Copyright © 2012 Accenture. All rights reserved.

64

Ferramentas de Instalação

Laboratório

- Desativar o sistema e selecionar a base de dados da redução III
- Partir o sistema e verificar se o sistema partiu corretamente

Copyright © 2012 Accenture. All rights reserved.

65

Ferramentas de Instalação

Configuração da Base de Dados de Demonstração

- A base de dados do runtime do sistema nos micros de controle é configurada com os drivers de dispositivos de hardware.
- Os drivers precisam ser substituídos por outros de simulação para rodar o sistema na plataforma de desenvolvimento.
- A configuração da base de dados para rodar como demonstração (simulação) é feita através do comando **InstalaDemo** do diretório /score/util conforme a sintaxe abaixo:
 - # InstalaDemo [-o]** ► o parâmetro -o configura os nodos 3 e 4 como nodos de operação

Copyright © 2012 Accenture. All rights reserved.

66

Ferramentas de Instalação

Laboratório

- Desativar o sistema e instalar um backup de produção da base de dados da redução III
- Partir o sistema e verificar o que acontece
- Executar o programa **InstalaDemo**
- Partir o sistema e verificar se funcionou normalmente

Copyright © 2012 Accenture. All rights reserved.

67

Ferramentas de Compilação

Compilação do Sistema

- A compilação do sistema é feita através do compilador **wcc** e as bibliotecas geradas pelo **wlib**. Entretanto, no Score esses recursos , assim como outros foram encapsulados em alguns comandos, que podem ser vistos abaixo:

- # GeraScore ► compila o sistema passando pelos diretórios. Se houver algum erro de sintaxe a compilação é interrompida
- # GeraTudo ► compila o sistema passando pelos diretórios. Se houver algum erro de sintaxe a compilação passa para o diretório seguinte
- # apaga_objs ► apaga todos arquivos objeto .o para forçar a compilação do sistema completo

Copyright © 2012 Accenture. All rights reserved.

68

Ferramentas de Instalação

Laboratório

- Compilar o sistema completo

Copyright © 2012 Accenture. All rights reserved.

69

Índice

Apresentação
Ferramentas de Desenvolvimento
Arquitetura do Sistema
Estruturas de Dados
Algoritmo de Controle
Criação de Opção de Operação
Criação de Relatórios
Preparação e Instalação de Upgrades
Projeto Final

Copyright © 2012 Accenture. All rights reserved.

70

Arquitetura do Sistema

Principais Tarefas do Kernel do Sistema

Tarefa	Função
AdminSemaf	Gerenciamento de semáforos de acesso à base de dados
BdScore	Leitura/gravação da base de dados via rede.
Drvsad	Driver de comunicação com a ATN 1.4
EnvRemota	Envia mensagens de comando para as remotas via rede Echelon
Eventos	Gerenciamento de eventos do sistema.
EventosMsg	Indicar os alarmes de eventos na linha de alarmes da IHM
IhmConsole	IHM da console do micro de controle.
Log	Gerenciamento de operações com arquivos de logs do módulo histórico
LogSon	Instanciada a cada log criado para gerenciar a atualização do respectivo arquivo de log
RecChaves	Reconhecer e tratar as sinalizações de mudança de chaves do painel da cuba
RecRemota	Receber as mensagens das remotas pela rede Echelon e envia-as ao driver da PCLTA
RedeAtn7	Driver de comunicação com a placa PCLTA
SalvaBd	Salvar a área comum, log, eventos e históricos da memória principal para HD e Hot-Standby
Score	Partir o sistema Score e criar suas principais tarefas residentes
TabRelGraf	Criar a estrutura de alarmes, gráficos e mini-gráficos o startup e gerenciar sua atualização a cada ciclo de controle
WatchCtrS	Verificar para o Hot-Standby se o controle está ativo no outro micro via cabo serial
Watchdog	Verificar a integridade do sistema e comandar a transferência para o Hot-Standby

Copyright © 2012 Accenture. All rights reserved.

71

Arquitetura do Sistema

Principais Tarefas de Controle de Processo

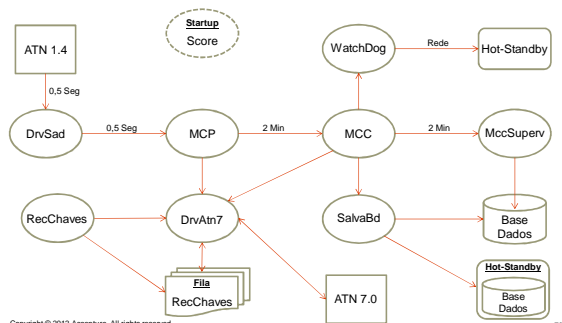
Tarefa	Função
Mcc	Implementar o ciclo de controle ajustando a resistência da cuba e ativando as demais tarefas desse ciclo a cada 2 minutos
Mcp	Implementar o ciclo de leitura, ativar as demais tarefas desse ciclo e a cada 240 ciclos de leitura (2 minutos) ativar o ciclo de controle
McpCorrida	Tratar a corrida de metal das cubas no ciclo de leitura.
TrataEa	Tratar o efeito anódico da cuba.

Copyright © 2012 Accenture. All rights reserved.

72

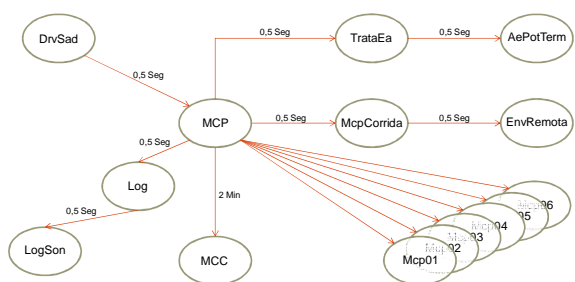
Arquitetura do Sistema

Diagrama de Comunicação – Visão Geral



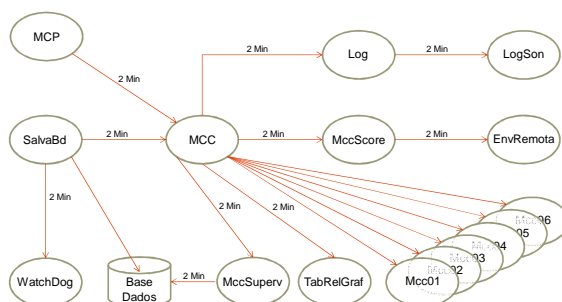
Arquitetura do Sistema

Diagrama do Ciclo de Leitura – MCP (0,5 Seg)



Arquitetura do Sistema

Diagrama do Ciclo de Controle – MCC (2 Min)



Arquitetura do Sistema

Ciclos de Execução do Algoritmo

- Ciclo de Leitura (MCP)
 - A cada 0,5 seg através da tarefa MCP
 - Ativada pelo DrvSad
 - Implementação dos cálculos primários
 - Ativa o MpcCorrida e mais 6 módulos residentes de ciclo de leitura
- Ciclo de Controle (MCC)
 - A cada 240 ciclos de leitura ou 2 min
 - Ativada pelo MCP
 - Implementa os cálculos de controle de resistência
 - Ativa várias tarefas e mais 6 módulos residentes de ciclo de controle

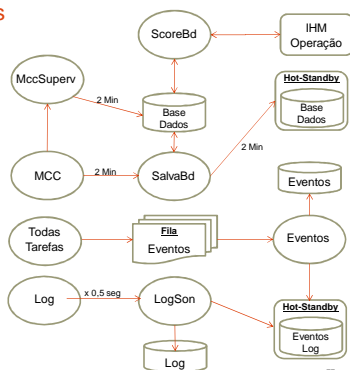
Copyright © 2012 Accenture. All rights reserved.

76

Arquitetura do Sistema

Base de Dados e Filas

- SalvaBd copia base de dados para hot-standby a cada 2 min
- Leitura e gravação via IHM Operação através do ScoreBd:
 - Config. Parâmetros
 - Opções de Operação
- MccSuperv trata dados históricos diários/turno
- Todas tarefas do sistema geram eventos
- Logs gravados a cada 2 min com base de tempo múltipla de 0,5 seg

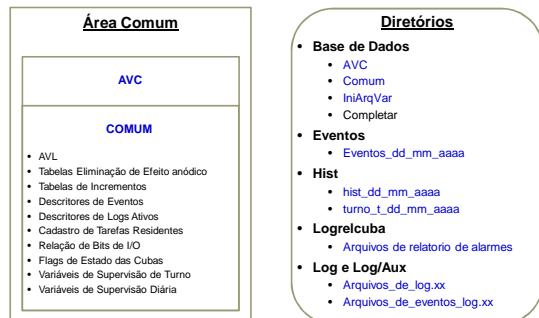


Copyright © 2012 Accenture. All rights reserved.

77

Arquitetura do Sistema

Estruturas de Dados e Diretórios



Copyright © 2012 Accenture. All rights reserved.

78

Arquitetura do Sistema

Laboratório

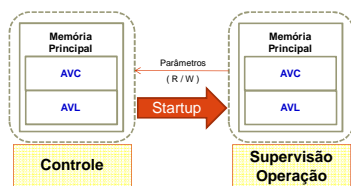
- Partir o sistema de controle na máquina virtual QNX_VM-1
- Partir o sistema de controle Hot-Standby na máquina virtual QNX_VM-2
- Acompanhar a atualização do Hot-Standby após o startup
- Verificar quais arquivos são atualizados em cada ciclo de controle depois que o Hot-Standby ficou pronto para assumir o sistema

Copyright © 2012 Accenture. All rights reserved.

79

Arquitetura do Sistema

Filosofia – Programas de Configuração



- **Startup:** Base de Dados copiada **Controle** → **Operação/Supervisão**
- Gravação de Parâmetros:
 1. **Base de Dados:** **Controle** → **Operação/Supervisão**
 2. Aloca Semáforo
 3. **Base de Dados:** **Operação/Supervisão** → **Controle**
 4. Libera Semáforo

Copyright © 2012 Accenture. All rights reserved.

80

Arquitetura do Sistema

Laboratório

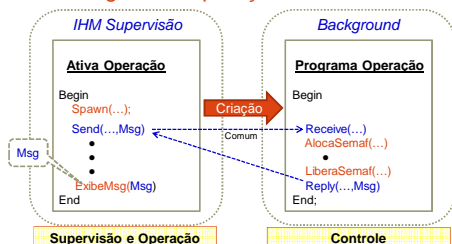
- Carregar o programa de configuração de parâmetros de cuba na IHM de supervisão
- Verificar pelo comando **sin** em qual nodo o programa de configuração está rodando

Copyright © 2012 Accenture. All rights reserved.

81

Arquitetura do Sistema

Filosofia – Programas Operação



- IHM Supervisão
 - Cria Tarefa via rede
 - Envia endereço comum
 - Exibe mensagem execução
- Nodo Controle
 - Executa programa operação
 - Atualiza base de dados
 - Retorna mensagem de execução

Copyright © 2012 Accenture. All rights reserved.

82

Arquitetura do Sistema

Laboratório

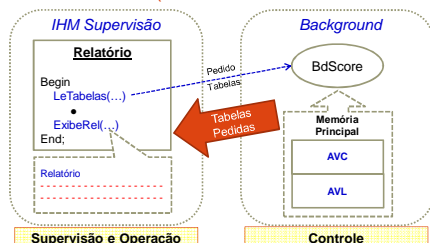
- Colocar um **sleep (40)** no início do programa exemplo de operação do toolkit, recompilar e atualizar no nodo de controle
- Executar o programa de operação
- Verificar pelo comando **sin** em qual nodo o programa de operação está rodando

Copyright © 2012 Accenture. All rights reserved.

83

Arquitetura do Sistema

Filosofia – Relatórios (Ciclo de Controle e Históricos)



- IHM Supervisão
 - Relatório atualiza tabelas
 - Processa e Formata saída
 - Exibe relatório
- Nodo Controle
 - Recebe pedido de tabelas
 - Atualiza tabelas pedidas

Copyright © 2012 Accenture. All rights reserved.

84

Arquitetura do Sistema

Laboratório

- Perguntas:
 - Uma tarefa residente do sistema pode atribuir valor a um parâmetro?
 - Preciso desativar o sistema de controle para instalar um programa de operação?
 - Que cuidados preciso tomar na leitura de dados do relatório?
- Executar o relatório de ciclo de controle:
 - Carregar um relatório de ciclo de controle na IHM de supervisão
 - Verificar pelo comando **sin** em qual nodo o relatório está rodando

Copyright © 2012 Accenture. All rights reserved.

85

Arquitetura do Sistema

Principais Diretórios do Score Runtime

Diretório	Conteúdo
/score/base_dados	Bases de dados do SCORE (avc, comum, etc...)
/score/bin	Comandos do SCORE.
/score/config	Configuração do IHM do SCORE e do ambiente QNX.
/score/erros	Registros de erros ocorridos no SCORE.
/score/eventos	Eventos históricos do SCORE.
/score/exec	Programas executáveis e descritores de relatórios.
/score/help	Textos de ajuda do IHM.
/score/hist	Históricos diários e de turno do SCORE.
/score/log	Logs gerados pelo módulo histórico.
/score/logrelcuba	Logs gerados pelo relatório de alarmes.
/score/telas	Telas da interface gráfica do SCORE
/score/testes	Programas de testes da remota ATN 7.0 e da interface analógica ATN 1.4
/score/tmp	Arquivos gerados temporariamente pelo SCORE
/score/util	Utilitários do SCORE.

Copyright © 2012 Accenture. All rights reserved.

86

Arquitetura do Sistema

Principais Diretórios de Fontes

Diretório	Conteúdo
/score/srcs/entdados	Bibliotecas de edição
/score/srcs/eventos	Prólogos de eventos
/score/srcs/ihm	Bibliotecas de ihm de console
/score/srcs/ihm_windows	Bibliotecas de ihm windows
/score/srcs/include	Prólogos dos programas.
/score/srcs/interfrem	Biblioteca de interface com a remota
/score/srcs/lib_ihm_ph	Biblioteca da ihm Photon
/score/srcs/lib_ph	Biblioteca da ihm Photon
/score/srcs/lib_score	Documentação da biblioteca de funções genéricas.
/score/srcs/mcc/MccCba	Algoritmo de ciclo de controle da CBA
/score/srcs/mcp/McpCbaTk	Algoritmo de ciclo de leitura da CBA
/score/srcs/ope	Exemplo de programas de operação.
/score/srcs/queue	Biblioteca, prólogo e detalhamento das funções do queue
/score/srcs/rel	Bibliotecas e exemplos de relatórios
/score/srcs/util	Programas fontes de utilitários do sistema.

Copyright © 2012 Accenture. All rights reserved.

87

Estrutura de Dados

Principais Prólogos do Sistema

Prólogo	Conteúdo
BdScore.h	Definições de acesso à base de dados do sistema.
Definicao.h	Operadores da linguagem C usados no sistema
DescrArqVar.h	Configuração das variáveis do SCORE para acesso pelo módulo histórico, módulo de configuração e opções de exportação.
Eventos.h	Códigos de eventos do sistema.
EventosUser.h	Códigos de eventos específicos do usuário.
EventosMsg.h	Definições de tipos do descritor de eventos.
IniArqVar.h	Definições das estruturas e tipos utilizados pelo DescrArqVar.h.
Listas.h	Protótipos das funções da biblioteca Listas.lib
Macros.h	Definições das macros utilizadas no sistema.
OperacaoBibW.h	Protótipos das funções da biblioteca ConsoleW.lib
Prototipos.h	Protótipos das funções da biblioteca Score.lib
Relatorios.h	Definições de estruturas de dados e constantes dos relatórios.
RelPrototipos.h	Protótipos das funções da biblioteca BibRel.lib
Score.h	Principais estruturas de dados dos arquivos avc e comum.
ScoreUser.h	Estruturas de dados do AVC e AVL específicos do usuário.
TiposOpw.h	Principais definições de constantes do sistema.
ScoreConst.h	Códigos de erros e suas respectivas mensagens.
ScoreErros.h	Definições da base de dados de strings para estrutura multi-lingua do sistema.
ScoreMsgs.h	Definições das estruturas de dados e constantes da IHM Windows.
VarsComum.h	Definições dos ponteiros das estruturas de dados do SCORE.

Copyright © 2012 Accenture. All rights reserved.

88

Arquitetura do Sistema

Laboratório

- Percorrer os diretórios de mcp, mcc, operação e relatórios
- Entrar no diretório include e editar os prólogos com os seguintes conteúdos
 - Declaração da área comum
 - Códigos de eventos
 - Macros de acesso ao AVC, AVL e variáveis históricas diárias e de turno
 - Protótipo das funções da biblioteca LibScore
- Abrir o arquivo .doc do diretório lib_score e verificar seu conteúdo
- Fazer um programa que exibe na tela o tamanho dos principais tipos de variáveis utilizados pelo sistema (byte, short, int, long, float, double)

Copyright © 2012 Accenture. All rights reserved.

89

Índice

Apresentação

Ferramentas de Desenvolvimento

Arquitetura do Sistema

Estruturas de Dados

Algoritmo de Controle

Criação de Opção de Operação

Criação de Relatórios

Preparação e Instalação de Upgrades

Projeto Final

Copyright © 2012 Accenture. All rights reserved.

90

Estrutura de Dados

Variáveis do Sistema

- Variáveis de Processo
 - Informações de processo:
 - **AVL** : Arquivo de Variáveis de Linha
 - **AVC** : Arquivo de Variáveis de Cuba
 - Classes de variáveis:
 - Parâmetros de configuração
 - Variáveis de trabalho
 - Flags de estado
 - Variáveis de supervisão diária/turno
- Comum
 - Descritor de Eventos
 - Tabelas de incrementos
 - Tabelas de supressão de efeito anódico
 - Lista de tarefas residentes
 - Relação de bits de I/O
 - Descritor de Logs Ativos

Copyright © 2012 Accenture. All rights reserved.

91

Estrutura de Dados

Passos de Criação de Variáveis no AVL e AVC

- Criação, Utilização e Instalação
 1. Declarar a variável na respectiva estrutura do prólogo **ScoreUser.h**
 2. Incluir a variável no arquivo **DescrArqVar.h** para ficar acessível pela IHM de supervisão do sistema
 3. Incluir o texto de ajuda da variável no arquivo de help no respectivo arquivo de configuração de linha ou cuba.
 4. Incluir a referência e atribuição da variável nos programas
 5. Recompilar todo o sistema Score e testar a nova versão da aplicação
 6. Preparar um upgrade do **Runtime** do sistema com os arquivos alterados, incluindo o arquivo **IniArqVar.dat** e instalar

Copyright © 2012 Accenture. All rights reserved.

92

Estrutura de Dados

Exemplo de Criação de Parâmetro AVC

- Diretrizes da declaração
 - Declaração na respectiva estrutura do prólogo **ScoreUser.h**
 - O tamanho da estrutura é fixo, por isso a posição das variáveis não pode ser deslocada.
 - **Exemplo:** criar variável **VincMax** tipo **float** (4 bytes) na 1ª tabela de parâmetros de cuba **t_par1_user**
- **1º Passo:** declarar a variável na estrutura **t_par1_user**
 - Estrutura antes da inclusão da variável:

```
volatile struct par1_user_avc /* 1a. tabela de parametros do usuario */
{
    /* Parametros de Usuario */
    float      Bemf;           /* Força contra eletromotriz */
    float      VSetPoint;      /* Tensao set-point */
    float      Livre1;         /* Variavel disponivel */
    float      EaVoltLim;      /* Tensão limite de EA */
    char       ResParUser[496]; /* reserva */
} t_par1_user;
```

Copyright © 2012 Accenture. All rights reserved.

93

Estrutura de Dados

Exemplo de Criação de Parâmetro AVC

- **1ª Opção** : declarar a variável no final da estrutura
 1. Incluir uma linha com a declaração da nova variável
 2. Deduzir o número de bytes da variável na reserva **ResParUser**
- Estrutura depois da inclusão da variável:


```
volatile struct par1_user_avc /* 1a. tabela de parametros do usuario */
{
    /* Parametros de Usuario */
    float      Bemf;           /* Força contra eletromotriz */
    float      VSetPoint;      /* Tensao set-point */
    float      Livre1;         /* Variavel disponivel */
    float      EaVoltLim;      /* Tensão limite de EA */
    float      VIncMax;        /* Tensão limite de incremento */
    char       ResParUser[492]; /* reserva 496 - 4 = 492 */
} t_par1_user;
```

Copyright © 2012 Accenture All rights reserved.

94

Estrutura de Dados

Exemplo de Criação de Parâmetro AVC

- **2ª Opção** : substituir uma variável que não é mais utilizadas
 1. Incluir uma linha com a declaração da nova variável
 2. Deduzir o número de bytes da variável na reserva **ResParUser**
- Estrutura depois da inclusão da variável:


```
volatile struct par1_user_avc /* 1a. tabela de parametros do usuario */
{
    /* Parametros de Usuario */
    float      Bemf;           /* Força contra eletromotriz */
    float      VSetPoint;      /* Tensao set-point */
    float      VIncMax;        /* Substituida Livre1 por VIncMax */
    float      EaVoltLim;      /* Tensão limite de EA */
    char       ResParUser[496]; /* reserva */
} t_par1_user;
```

Copyright © 2012 Accenture All rights reserved.

95

Estrutura de Dados

Exemplo de Criação de Parâmetro AVC

- **2º Passo**: incluir a variável no arquivo **DescrArqVar.h**
- Cada linha refere a uma variável configurada pelos parâmetros do quadro

Campo	Tipo	Descrição
NomeVar	ascii	Nome do parâmetro
UeVar	ascii	Unidade de engenharia
EndrVar	ponteiro	Endereço do parâmetro
NivelSenha	define	Nível de senha do parâmetro
TipoVar	define	Tipo do parâmetro
FmtVar	ascii	Formato do parâmetro
LimInf	real	Limite Inferior
LimSup	real	Limite Superior
FatConv	real	Fator de conversão para unidade desejada
Classe	define	Classe do parâmetro
FlagAlg	define	Algoritmo ao qual o parâmetro pertence
Rotulo	ascii	Rótulo do parâmetro no relatório de alarmes
Tag	ascii	Tag do parâmetro referido no relatório de alarmes
GrupoVar	define	Grupo do parâmetro

- O domínio de valores está declarado no arquivo **IniArqVar.h**

Copyright © 2012 Accenture All rights reserved.

96

Estrutura de Dados

Exemplo de Criação de Parâmetro AVC

- A variável no **DescrArqVar.h** deve ser declarada no grupo correspondente à estrutura do **ScoreUser.h** na qual foi declarada, conforme quadro ao lado.

Assim, a variável **VincMax** declarada em **t_par1_user** no **ScoreUser.h** será configurada em **VarPar1** no **DescrArqVar.h**

DescrArqVar.h	Score.h
VarParamAvc	t_param_avc
VarSuperv	t_avc_supervisao
VarEst	t_est_avc
VarAvl	t_avl
VarPar1	t_par1_user
VarPar2	t_par2_user
VarUser1	t_user1_avc
VarUser2	t_user2_avc
VarUser3	t_user3_avc
VarUser4	t_user4_avc

- **3º Passo:** incluir o texto de ajuda no arquivo **ParCubasW.hlp**
 - O nome da variável entre [] funciona como uma chave de busca para acessar a descrição
 - O tamanho do texto de help é fixo e limitado a 5 linhas.

Copyright © 2012 Accenture. All rights reserved.

97

Estrutura de Dados

Exemplo de Criação de Parâmetro AVC

- **4º Passo:** incluir a referência da variável nos programas

- Declaração da estrutura de parâmetros:

```
volatile struct arq_var_cubas /* Arquivo de variáveis das cubas */
{
    •
    t_par1_user      ParUser1[MAX_CUBAS];
    t_par2_user      ParUser2[MAX_CUBAS];
    •
} t_avc;
```

- Parâmetro declarado em **t_par1_user**:

```
If (AVC.User1(Cuba).CCicCont.Vinc > AVC.Par1User[Cuba].VincMax)
AVC.User1(Cuba).CCicCont.Vinc = AVC.Par1User[Cuba].VincMax;
```
- Parâmetro declarado em **t_par2_user**:

```
If (AVC.User1(Cuba).CCicCont.Vinc > AVC.Par2User[Cuba].VincMax)
AVC.User1(Cuba).CCicCont.Vinc = AVC.Par2User[Cuba].VincMax;
```

Copyright © 2012 Accenture. All rights reserved.

98

Estrutura de Dados

Exemplo de Criação de Parâmetro AVC

- **5º Passo:** recompilar todo o sistema Score e testar a nova versão da aplicação na plataforma de desenvolvimento
GeraScore
- **6º Passo:** preparar e instalar o upgrade de **Runtime** incluindo dentre os arquivos alterados o **ArqVar.dat** e **ArqVar.dat** do diretório **/score/base_dados**.
 - A instalação dos arquivos arqvar.dat deve ser feita utilizando setup devido à diferença entre a ATN7 e ATN8.
- **Observação:** esse exemplo vale para qualquer estrutura de dados da estrutura AVC e AVL do prólogo **ScoreUser.h**, desde que aplicado na respectiva estrutura de dados.

Copyright © 2012 Accenture. All rights reserved.

99

Estrutura de Dados

Roteiro de Criação de Variáveis de Linha (AVL)

• Criação, Utilização e Instalação (Exemplo AVC)

1. Declaração da variável na estrutura adequada do prólogo **ScoreUser.h** em **t_avl** :
 - **t_par_avl_usu** ▶ **Par** : parâmetro de configuração
 - **t_trab_avl_usu** ▶ **Trab** : variável de trabalho
 - **t_sup_avl_usu** ▶ **Sup** : variável de supervisão diária/turno
2. Incluir a variável no arquivo **DescrArqVar.h** para ficar acessível pela opção de **Configuração de Parâmetros de Linha** da IHM (se necessário).
3. Incluir texto de ajuda da variável no arquivo **ParLinhaW.hlp** (se necessário)
4. Incluir a referência da variável **NomeVar** nos programas
 - **AVL.Par.NomeVar** : parâmetro de configuração
 - **AVL.Trab.NomeVar** : variável de trabalho
 - **AVL.Sup.NomeVar** : variável de supervisão diária
 - **AVL.TURNO.Sup.NomeVar** : variável de supervisão de turno
5. Recompilar todo o sistema Score e testar a nova versão da aplicação
6. Gerar e instalar a nova versão do sistema e incluindo arquivo **IniArqVar.dat**

Copyright © 2012 Accenture. All rights reserved.

100

Estrutura de Dados

Roteiro de Criação de Variáveis de Cubas (AVC)

• Criação, Utilização e Instalação (Exemplo AVC)

1. Declaração da variável na estrutura adequada do prólogo **ScoreUser.h** em **t_avc** :
 - **t_par1_user** ▶ **Par1User**[MAX_CUBAS] : 1ª tabela de parâmetros
 - **t_par2_user** ▶ **Par2User**[MAX_CUBAS] : 2ª tabela de parâmetros
 - **t_user1_avc** ▶ **User1**[MAX_CUBAS] : 1ª tabela de variáveis de trabalho
 - **t_user2_avc** ▶ **User2**[MAX_CUBAS] : 2ª tabela de variáveis de trabalho
 - **t_user3_avc** ▶ **User3**[MAX_CUBAS] : 3ª tabela de variáveis de trabalho
 - **t_user4_avc** ▶ **User4**[MAX_CUBAS] : 4ª tabela de variáveis de trabalho
 - **t_est_usu** ▶ **Est** : variável de status
 - **t_sup_avc_usu** ▶ **Sup** : variável de supervisão diária/turno

Copyright © 2012 Accenture. All rights reserved.

101

Estrutura de Dados

Roteiro de Criação de Variáveis de Cubas (AVC)

• Criação, Utilização e Instalação (Exemplo AVC)

2. Incluir a variável no arquivo **DescrArqVar.h**, possibilitando o acesso através da opção de **Configuração de Parâmetros de Cuba** da IHM.
3. Incluir o texto de ajuda do parâmetro (somente) no arquivo **ParCubasW.hlp**

Copyright © 2012 Accenture. All rights reserved.

102

Estrutura de Dados

Roteiro de Criação de Variáveis de Cubas (AVC)

• Criação, Utilização e Instalação (Exemplo AVC)

4. Incluir a referência da variável **NomeVar** nos programas

- **AVC.Par1User[Cuba].NomeVar** : 1ª tabela de parâmetros
- **AVC.Par2User[Cuba].NomeVar** : 2ª tabela de parâmetros
- **AVC.User1[Cuba].NomeVar** : 1ª tabela de variáveis de trabalho
- **AVC.User2[Cuba].NomeVar** : 2ª tabela de variáveis de trabalho
- **AVC.User3[Cuba].NomeVar** : 3ª tabela de variáveis de trabalho
- **AVC.User4[Cuba].NomeVar** : 4ª tabela de variáveis de trabalho
- **EST_AVC(Cuba).Est.NomeVar** : variável de status
- **SUPERV_AVC(Cuba).Sup.NomeVar** : variável de status
- **AVL.Sup.NomeVar** : variável de supervisão diária
- **SUPERV_AVC_TURNO(Cuba).Sup.NomeVar** : variável de supervisão de turno

Copyright © 2012 Accenture. All rights reserved.

103

Estrutura de Dados

Roteiro de Criação de Variáveis de Cubas (AVC)

• Criação, Utilização e Instalação (Exemplo AVC)

5. Recompilar todo o sistema Score e testar a nova versão da aplicação
6. Preparar um upgrade do *Runtime* do sistema com os arquivos alterados, incluindo o arquivo **IniArqVar.dat** e instalar

Copyright © 2012 Accenture. All rights reserved.

104

Arquitetura do Sistema

Laboratório

• Criar as seguintes variáveis na base de dados:

- **ParLTeste**: parâmetro de linha (int)
- **VarCTeste**: variável de cuba (float)
- **FlagCTeste** : flag de status (byte)

• Exibir as variáveis criadas de acordo com os seguintes critérios:

- **ParLTeste**: somente no programa de configuração de linha, não pode entrar no relatório de parâmetros
- **VarCTeste**: programa de configuração e relatório de parâmetros de cuba
- **FlagCTeste** : relatório de parâmetros de cuba e na área de status do relatório de alarmes de cuba

Copyright © 2012 Accenture. All rights reserved.

105

Índice

Apresentação
 Ferramentas de Desenvolvimento
 Arquitetura do Sistema
 Estruturas de Dados
Algoritmo de Controle
 Criação de Opção de Operação
 Criação de Relatórios
 Preparação e Instalação de Upgrades
 Projeto Final

Copyright © 2012 Accenture. All rights reserved.

106

Algoritmo de Controle

Introdução

- Ciclos de execução
 - **Ciclo de Leitura:** MCP – Módulo de Cálculos Primários
 - **Ciclo de Controle:** MCC – Módulo de Ciclo de Controle
- Implementação dos algoritmos
 - Arquitetura das tarefas (MCP e MCC)
 - Implementação das tarefas
 - Compilação e depuração da tarefa
 - Configuração do startup da tarefa na base de dados
- Criação de eventos
 - Declaração do código do evento no prólogo
 - Configuração dos parâmetros na base de dados
 - Geração do evento no programa

Copyright © 2012 Accenture. All rights reserved.

107

Algoritmo de Controle

MCP e MCC - Implementação da Tarefa

- Esboço do Algoritmo
 - Declaração de variáveis e estruturas de dados
 - Algoritmo da tarefa (lógicas de programação, comunicação entre tarefas, funções, etc...)
- Declaração de Variáveis

Início

```
const INICIALIZACAO=1
const CICLO_CONTROLE=2
const CICLO_LEITURA=3
const DESABILITOU_LINHA=4
tipo MsgSinc = registro
    inteiro : Origem;
    inteiro : Acao;
    tipo v : arranjo [1:8] inteiro;
fim registro
v : Parametro;
MsgSinc : Msg;
```

Copyright © 2012 Accenture. All rights reserved.

108

Algoritmo de Controle

MCP - Implementação da Tarefa

• Algoritmo do MCP

- Receber mensagem com endereço da área comum (SCORE).
- Inicializar ponteiros para área comum.
- Fazer consistência dos parâmetros da chamada da tarefa.
- enquanto (VERDADEIRO) faça
 - Receber mensagem de ativação do DRVSAD (0,5 seg)
 - Pegar hora atual
 - escolha (Msg.Acao)
 - caso
 - INICIALIZACAO :
 - Executar procedimentos de inicialização.
 - Aguardar partida do sistema se Hot-Standby.
 - caso
 - CICLO_LEITURA :
 - ProcessaCicloLeitura(Cubainicial, CubaFinal)
 - caso
 - DESABILITOU_LINHA :
 - Tratar desabilitação da linha.
 - fim escolha
 - Enviar resposta a mensagem de ativação
- fim enquanto
- Fim

Copyright © 2012 Accenture. All rights reserved.

109

Algoritmo de Controle

MCP - Implementação da Tarefa

- Cuidados na Implementação
 - O MCP executa um ciclo de leitura a cada 0,5 segundo, portanto:
 - Qualquer comando que aguarde confirmação ou coloque a tarefa em espera **desativa o sistema**
 - A tempo de execução da tarefa sempre deve ser inferior a 0,5 seg, caso contrário o sistema será desativado.
- Ambiente de Desenvolvimento:
 - **Diretório MCP:** `/score/srcs/mcp`
 - **Exemplos:** `McpUser.h` e `McpUser.c`
 - **Makefile:** `makeuser`
- O diretório `/score/srcs/mcp/McpCbaTk` contém o algoritmo de ciclo de leitura que está em produção na CBA

Copyright © 2012 Accenture. All rights reserved.

110

Arquitetura do Sistema

Laboratório

- Abrir o diretório `/score/srcs/mcp` e analisar seus principais arquivos

Copyright © 2012 Accenture. All rights reserved.

111

Algoritmo de Controle

MCC - Implementação da Tarefa

- Algoritmo do MCC
 - Receber mensagem com endereço da área comum (SCORE).
 - Inicializar ponteiros para área comum.
 - Fazer consistência dos parâmetros da chamada da tarefa.
 - enquanto (VERDADEIRO) faça
 - Receber mensagem de ativação MCP (2 min)
 - Pegar hora atual
 - se (Msg.Origem = MCC)
 - escolha (Msg.Acao)
 - caso INICIALIZACAO :
 - Executar procedimentos de inicialização.
 - Aguardar partida do sistema se Hot-Standby.
 - caso CICLO_LEITURA :
 - ProcessaCicloControle(CubaInicial, CubaFinal)
 - caso DESABILITOU_LINHA :
 - Tratar desabilitação da linha.
 - fim escolha
 - fim se
 - Enviar resposta a mensagem de ativação
 - fim enquanto

Copyright © 2012 Accenture. All rights reserved.

112

Algoritmo de Controle

MCC - Implementação da Tarefa

- Cuidados na Implementação
 - O MCC executa um ciclo de controle a cada 2 minutos, portanto:
 - Qualquer comando que aguarde confirmação ou coloque a tarefa em espera **desativa o sistema** após 2 minutos
 - A tempo de execução da tarefa sempre deve ser inferior a 2 minutos, caso contrário o sistema será desativado.
- Ambiente de Desenvolvimento:
 - **Diretório MCP:** /score/srcs/mcc/MccCba
 - **Exemplos:** MccUser.h e MccUser.c
 - **Makefile:** makeuser
- O diretório /score/srcs/mcp/MccCba contém o algoritmo de ciclo de controle que está em produção na CBA

Copyright © 2012 Accenture. All rights reserved.

113

Arquitetura do Sistema

Laboratório

- Abrir o diretório /score/srcs/mcc e analisar seus principais arquivos

Copyright © 2012 Accenture. All rights reserved.

114

Algoritmo de Controle

Implementação, Compilação e Depuração

- Detalhes de implementação
 - **Editor de textos:** [vedit](#)
 - Linguagem e Implementação
 - Programa em linguagem C, respeitando sua sintaxe e padrões
 - Toda tarefa [.c](#) tem respectivo prólogo [.h](#), ex: [MccUser.c](#) e [MccUser.h](#)
 - Compilação:
 - **Compilador:** Watcom C/C++ versão 10.6
 - Relações de dependências declaradas no **makefile**
 - Depuração:
 - **Comando:** **wd**
 - **Diretiva de Compilação:** **-g** (macro DEBUG)
 - Instalação:
 - Instalação de nova tarefa residente na base de dados via [InstalaCtrl](#) visto no capítulo [Ferramentas de Desenvolvimento](#)

Copyright © 2012 Accenture. All rights reserved.

115

Algoritmo de Controle

Procedimento de Criação de uma Tarefa Residente

- No exemplo seguinte utilizaremos como exemplo uma tarefa de ciclo de leitura [McpUser](#), mas lembramos que o procedimento é o mesmo tanto para tarefas de ciclo de leitura e controle.
- Destacamos que esse procedimento deve ser realizado na plataforma de desenvolvimento.
- **1º Passo** : implementar os arquivos fonte [McpUser.h](#) e [McpUser.c](#)
- **2º Passo** : criar variáveis no [ScoreUser.h](#) e configurar no [DescrArqVar.h](#)
- **3º Passo** : incluir as dependências no [makefile](#) e compilar:


```
# cd /score/srcs
# GeraScore
```
- **4º Passo** : instalar a tarefa na base de dados de simulação pelo [InstalaCtrl](#) visto no capítulo [Ferramentas de Desenvolvimento](#)

Copyright © 2012 Accenture. All rights reserved.

116

Criação de Relatórios

Procedimento de Criação de uma Tarefa Residente

- **5º Passo** : depurar e testar a tarefa
- **6º Passo** : preparar um upgrade com os arquivos alterados conforme capítulo [Preparação e Instalação de Upgrades](#)
 - O procedimento de instalação da partida da tarefa via [InstalaCtrl](#) deve ser repetido na instalação do 1º micro de controle da redução com o sistema desativado.

Copyright © 2012 Accenture. All rights reserved.

117

Algoritmo de Controle

Depuração da Tarefa

- Utilitário dumper gera um arquivo de erro **.dmp** no diretório **/score/erros** quando um programa morre por erro de execução
- Usando como exemplo o programa **McpUser**, o arquivo é acessado pelo comando **wd** conforme a sintaxe abaixo:
 - # **Wd -trap=pmd /score/erros/McpUser.dmp**
 - Depurador entra direto no ponto que o programa morreu
 - **Show → Calls**: mostra, em alguns casos, a sequência de funções chamadas antes do erro.
 - Pode retornar às condições da tarefa antes do erro falha para identificar seu motivo, mostrando por exemplo os valores de variáveis.
- Requisitos da depuração utilizando **.dmp**
 - # **Dumper &**
 - # Programa compilado com diretiva **-g** (macro **DEBUG**)

Copyright © 2012 Accenture. All rights reserved.

118

Algoritmo de Controle

Laboratório

- Editar a tarefa Mcp e incluir uma divisão por zero
- Compilar e colocar em execução
- Abrir o arquivo **dmp** pelo depurador e verificar o status das variáveis antes da tarefa morrer
- Voltar o sistema à condição normal

Copyright © 2012 Accenture. All rights reserved.

119

Algoritmo de Controle

Pontos de Atenção

- Do mesmo modo que no micro de controle, o Hot-Standby também carrega todas as tarefas residentes em memória na sua partida. Portanto, quando uma tarefa residente modificada é instalada no Hot-Standby, é necessário sua desativação e reativação carregar a nova versão.
- Qualquer modificação feita no sistema deve ser sempre testada no micro de simulação.
- Se a modificação for muito grande e crítica, é seguro instalá-la primeiro no Hot-Standby, desativando-o e ativando-o novamente. Depois que já está pronto para assumir o controle, transfere-se o controle para ele e mantém o micro que estava como principal desativado na mesma condição que ele ficou para preservar sua base de dados. Somente depois que está garantido que a modificação não vai dar problema é que o outro micro é atualizado e entra como Hot-Standby.

Copyright © 2012 Accenture. All rights reserved.

120

Algoritmo de Controle

Criação de Opção de Algoritmo

- **1º Passo** : as opções de algoritmo são criadas na estrutura `t_NomeAlgCtr` do arquivo `DescrArqVar.h`, definido entre aspas com tamanho máximo de 8 caracteres, como foi criado `AlgUser1` no exemplo abaixo:

```
/* algoritmos do usuario */
t_NomeAlgCtr NAlgCtr[] =
{
    /* 09 */ "AlgUser1"
    /* 10 */ " "
    /* 11 */ " "
    /* 12 */ " "
    /* 13 */ " "
};
```

- **2º Passo** : Recompilar o arquivo `/score/util/IniArqvar` e instalar os arquivos `Arqvar.dat` e `Arqvar1.dat` na base de dados do sistema, exibindo o novo algoritmo na tela de pré-seleção.
- **3º Passo** : incluir a referência para o algoritmo nas tarefas.

Copyright © 2012 Accenture. All rights reserved.

121

Algoritmo de Controle

Criação de Eventos do Sistema

- Características dos Eventos
 - Registro de ocorrências importantes do sistema
 - Componentes do evento:
 - Cuba ou Faixa de Cubas
 - Mnemônico
 - Mensagem
 - 5 Parâmetros variáveis
- Resumo da Criação de Eventos
 1. Declaração do código do evento no prólogo
 2. Configuração dos parâmetros do evento no descritor
 3. Atualização da configuração do descritor no comum
 4. Atualização das telas de seleção de filtros de eventos
 5. Inclusão das informações de eventos no arquivo de *help*
 6. Inclusão da função de geração do evento no programa

Copyright © 2012 Accenture. All rights reserved.

122

Algoritmo de Controle

Criação de Eventos do Sistema

- **1º Passo** : Declarar o Código do Evento no Prólogo
 - Os códigos de eventos são declarados no arquivo `EventosCBA.h` do diretório `/score/srcs/include`
 - Sintaxe da declaração e parâmetros


```
# Define COD_EV      NumEv      /* DescrEv */
```

 - Descrição dos parâmetros:
 - **COD_EV**: nome da constante do evento que será utilizada como referência no código do programa
 - **NumEv**: valor numérico de 1 a 200, onde os valores de 1 a 74 são reservados para o Score e as novas aplicações podem utilizar a faixa de 75 a 200
 - **DescrEv**: descrição sucinta do evento criado

Copyright © 2012 Accenture. All rights reserved.

123

Algoritmo de Controle

Criação de Eventos do Sistema

- **2º Passo** : configurar os parâmetros do evento no descritor **DescrEvph** visto no capítulo *Ferramentas de Configuração*
- **3º Passo** : Atualizar a configuração do descritor de eventos no arquivo comum da base de dados através do utilitário **IniDescrEv** visto no capítulo *Ferramentas de Configuração*
- **4º Passo** : Incluir as Informações do Evento no Help:
 - Incluir o *mneumônico*, descrição e parâmetros do evento no arquivo de **HelpEv.hlp** do diretório **/score/help**

Copyright © 2012 Accenture. All rights reserved.

124

Algoritmo de Controle

Criação de Eventos do Sistema

- **6º Passo** : Incluir a Função de Geração do Evento nos Programas
 - O evento é gerado nos programas através da função **GeraEvento** da biblioteca **LibScore.lib**, conforme sintaxe no quadro abaixo:

Função	
GeraEvento(CodEv, CubaInicial, CubaFinal, Par1, Par2, Par3, Par4, Par5)	
Parâmetros de Chamada	
Parâmetros	Descrição
CodEv	Código do evento (define) declarado no prólogo EventosCBA.h
CubaInicial	Índice da cuba inicial na base de dados do SCORE. Colocar valor -1 quando não existir (evento de linha).
CubaFinal	Índice da cuba final na base de dados do SCORE. Colocar valor -1 quando não existir.
Par1 .. Par5	1º ao 5º parâmetro do evento, tipo variável de acordo com o evento

Copyright © 2012 Accenture. All rights reserved.

125

Algoritmo de Controle

Pontos de Atenção

- Os parâmetros passados para a função **GeraEvento** devem ser do mesmo tipo definido através do descritor de eventos. Erros nessa passagem de parâmetros podem provocar exceção no módulo e consequentemente a desativação do sistema.
- **Exemplo** :
 - Caso um parâmetro definido como **string** seja passado uma variável tipo **int** ou **float**, o valor da variável será utilizado como ponteiro para **string** com grande probabilidade de gerar exceção de memória.

Copyright © 2012 Accenture. All rights reserved.

126

Algoritmo de Controle

Laboratório

- Criar o evento **EvTeste** e com a mesma máscara do gerado pelo programa de operação de exemplo
- Substituir o evento no programa de operação
- Preparar o help do evento
- Instalar o evento e verificar se aparece na lista de filtros do relatório de eventos
- Executar o programa de operação e verificar
 - Se o evento foi gerado
 - Se o help do evento apareceu no relatório

Copyright © 2012 Accenture. All rights reserved.

127

Índice

Apresentação

Ferramentas de Desenvolvimento

Arquitetura do Sistema

Estruturas de Dados

Algoritmo de Controle

Criação de Opção de Operação

Criação de Relatórios

Preparação e Instalação de Upgrades

Projeto Final

Copyright © 2012 Accenture. All rights reserved.

128

Criação de Opção de Operação

Características do Programa de Operação

- Arquitetura padrão :
 - Criados pelo programa **AtivaOperação** pela IHM de supervisão
 - Rodam em background nos micros de controle

Algoritmo do Programa

- Receber mensagem de inicialização do **AtivaOperacao**.
- Pegar endereço da área comum.
- Fazer consistência dos parâmetros de chamada (IHM)
- se** (Parâmetro de Chamada Ok)
 - então**
 - para** Cubainicial **até** CubaFinal **faca**
 - Executar procedimentos específicos da opção de operação.
 - Gerar evento específico da operação executada.
 - Gerar evento de log da operação executada.
 - fim para**
- fim se**
- Retornar mensagem para o **AtivaOperacao** com status de execução

Copyright © 2012 Accenture. All rights reserved.

129

Criação de Opção de Operação

Procedimento de Criação da Opção de Operação

- **1º Passo** : implementar o programa com as funcionalidades do algoritmo
 - Padrão e sintaxe de programação da linguagem C
 - Editor de textos: **vedit**
 - Exemplo de programa de operação : **/score/srcs/ope/cba/OpeExemplo.c**
- **2º Passo** : compilar e gerar o módulo executável do programa
 - Incluir as dependências de compilação/linkedição do programa no **makefile** e compilar


```
# cd /score/srcs
# GeraScore
```
- **3º Passo** : depurar e testar o programa

Copyright © 2012 Accenture All rights reserved.

130

Criação de Opção de Operação

Procedimento de Criação da Opção de Operação

- **4º Passo** : configurar a chamada do programa na IHM respeitando as seguintes premissas:
 - O **Tag** do botão da tela e o campo **Tag** na configuração da chamada na IHM devem ter o mesmo nome do programa de operação a ser acionado pelo **AtivaOperacao**.
 - O campo **Nome** deve ser preenchido sempre com o **path** **/score/exec/AtivaOperacao**
 - O 1º argumento na configuração da IHM deve sempre ser do tipo **Tag**, indicando que nessa posição está sendo passado para o **AtivaOperacao** o nome do programa que ele criará no micro de controle via rede.
- **5º Passo** : instalar o programa de operação:
 - O programa de operação é carregado toda vez que acionado pela IHM, portanto não precisa desativar o sistema para atualizá-los.
 - Esses programas são instalados no diretório **/score/exec** como os demais

Copyright © 2012 Accenture All rights reserved.

131

Criação de Opção de Operação

Depuração de um Programa de Operação

Preparação do Programa para Depuração

- Incluir no makefile a diretiva **-g** (macro **DEBUG**)
- Preparar o programa para depuração pelo **wd**
 - Incluir o comando **sleep(30)** logo depois de receber a mensagem de sinalização do **AtivaOperacao**
 - Inibir os comandos de alocação e liberação de semáforos
- Compilar e gerar o módulo executável do programa alterado com a diretiva **-g**
- Depurar o programa no ambiente de desenvolvimento com o sistema de controle e interface gráfica rodando

Copyright © 2012 Accenture All rights reserved.

132

Criação de Opção de Operação

Procedimento de Criação da Opção de Operação

Depuração do Programa em Execução

- Partir o sistema de controle e interface gráfica
- Chamar o programa pela tela de operações da interface gráfica
- Pegar seu *pid* do processo através do comando **sin** imediatamente à ativação. Considerando o nome do processo começado com **Ope**, o comando deve ser executado numa console livre conforme a sintaxe abaixo :

```
# sin -P Ope
```
- Pegar o programa com o depurador **wd** passando como parâmetro o *pid* **PidOpe** conforme a sintaxe abaixo :

```
# wd PidOpe
```
- Esse comando **wd** deve ser executado num intervalo de tempo inferior a 30 segundos (**sleep**). Depois de passado esse intervalo o programa é liberado para depuração passo-a-passo dentro do **wd**.

Copyright © 2012 Accenture. All rights reserved.

133

Criação de Opção de Operação

Procedimento de Criação da Opção de Operação

Depuração do Programa Terminado por Erro de Execução

- Utilitário **dumper**
 - Gera um arquivo de depuração do programa **OpeExemplo** de *path* **/score/erros/OpeExemplo.dmp**
 - Deve ter sido ativado anteriormente conforme a sintaxe abaixo :

```
# dumper &
```
- Quando programa **OpeExemplo** morre por erro de execução seu arquivo de dump é acessado pelo depurador conforme a sintaxe abaixo

```
# wd -trap=pmd /score/erros/OpeExemplo.dmp
```
- O depurador entra direto no ponto que ocorreu o erro
- Selecionando as opções **Show → Calls** em alguns casos mostra a sequência de chamada de funções antes do erro, permitindo voltar à condição anterior e analisar as variáveis.

Copyright © 2012 Accenture. All rights reserved.

134

Criação de Opção de Operação

Laboratório

- Abrir o arquivo fonte de incrementos de operação
 - Abrir o arquivo fonte
 - Abrir a configuração dessa opção na IHM
- Analisar alguns tópicos da interface de chamada desse programa:
 - Passagem de parâmetros
 - Tratamento de opções de lista
 - Tratamento de strings
 - Tratamento e duração e tempo

Copyright © 2012 Accenture. All rights reserved.

135

Índice

Apresentação
 Ferramentas de Desenvolvimento
 Arquitetura do Sistema
 Estruturas de Dados
 Algoritmo de Controle
 Criação de Opção de Operação
Criação de Relatórios
 Preparação e Instalação de Upgrades
 Projeto Final

Copyright © 2012 Accenture. All rights reserved.

136

Criação de Relatórios

Características dos Relatórios

- Relatórios de Ciclo de Controle
 - Arquivos fonte do relatório **RelExemploCC**:
 - Prólogo RelExemploCC.h** : com a declaração das estruturas de dados de cabeçalho, área de dados e rodapé
 - Programa RelExemploCC.c** : com o código fonte do programa
 - Relatório de ciclo de controle:
 - Código Executável: RelExemploCC**
 - Descritor RelExemploCC.dsc** : máscara do relatório
 - Help RelExemploCC.hlp** : arquivo com descrição dos campos do relatório
 - Linkedição com múltiplas bibliotecas gera automaticamente a versão para impressora e tela do relatório

Copyright © 2012 Accenture. All rights reserved.

137

Criação de Relatórios

Características dos Relatórios

- Relatórios de Supervisão Diários e Turno
 - Arquivos fonte do relatório **RelExemploDia**:
 - Prólogo RelExemploDia.h** : declaração das estruturas de dados de cabeçalho, área de dados e rodapé
 - Programa RelExemploDia.c** : código fonte do programa
 - Códigos Executáveis**:
 - RelExemploDia** (diário)
 - RelExemploTDia** (turno)
 - Descritores**: máscara do relatório
 - RelExemploDia.dsc** (diário)
 - RelExemploTDia.dsc** (turno)
 - Help** : descrição dos campos do relatório
 - RelExemploDia.dsc** (diário)
 - RelExemploTDia.dsc** (turno)
 - Linkedição com múltiplas bibliotecas gera automaticamente a versão de impressora e tela dos relatórios diário e turno

Copyright © 2012 Accenture. All rights reserved.

138

Criação de Relatórios

Estruturas de Dados dos Relatórios

- Lay-Out do divido em 3 áreas: *cabeçalho*, *área de dados* e *rodapé*
- Todos relatórios têm um prólogo com extensão ".h" com mesmo nome do fonte ".c", onde são declaradas as estruturas dados:
 - Declaração do *cabeçalho*:

```
/*-- Cabeçalho --*/
typedef
struct VarCabece
{
    char      Selecao[14];
    .
    .
} t_cabec;
```

Copyright © 2012 Accenture. All rights reserved.

139

Criação de Relatórios

Estruturas de Dados dos Relatórios

- Declaração da *área de dados*:

```
/*-- Area de Dados --*/
typedef
struct VarDados
{
    int      NumCuba;
    float    Vincr;
    .
    .
} t_dados;
```

- Declaração do rodapé:

```
/*-- Rodape --*/
typedef
struct VarRodape
{
    float    VincrMed;
    .
    .
} t_rodape;
```

Copyright © 2012 Accenture. All rights reserved.

140

Criação de Relatórios

Estruturas de Dados dos Relatórios

- Declaração das estruturas no programa fonte ".c":

```
/*-- Definição de Variáveis Locais --*/
t_cabec      Cab;
t_dados      AreaDados[MAX_LINHAS];
t_rodape     Rodape;
```

Copyright © 2012 Accenture. All rights reserved.

141

Criação de Relatórios

Algoritmo dos Relatórios

Programa Principal

```

inteiro:      NLinhas;
t_cabecalho: Cab;
t_dados:    AreaDados[MAX_LINHAS];
t_rodape:   Rod;

- Inicializar o ambiente gráfico.
- Pegar endereço da área comum.
- Ler tabelas de dados atualizadas do micro de controle.
se ((Leitura de Tabelas Ok)
  então
    - Fazer consistência dos parâmetros de chamada.
    se (Parâmetros de Chamada Ok)
      então
        se ((NLinhas = GeraDadosRel(Cab,AreaDados,Rod)) > 0)
          então
            - Gerar o arquivo de dados do relatório.
            - Exibir as informações do relatório.
          fim se
        fim se
      fim se
    - Finalizar ambiente gráfico
  fim se

```

Copyright © 2012 Accenture. All rights reserved.

142

Criação de Relatórios

Algoritmo dos Relatórios

- Relatório de Ciclo de Controle

Função GeraDadosRel

```

inteiro:      NLinhas;
t_cabecalho: Cab;
t_dados:    AreaDados[MAX_LINHAS];
t_rodape:   Rod;

- Montar cabeçalho do relatório.
para Cubainicial ate CubaFinal faca
  - Montar área de dados da cuba.
  NLinhas = NLinhas + 1;
fim para

- Montar rodapé do relatório.
GeraDadosRel = NLinhas;

```

Copyright © 2012 Accenture. All rights reserved.

143

Criação de Relatórios

Algoritmo dos Relatórios

- Relatório de Supervisão Diária e Turno

Função GeraDadosRel

```

inteiro:      NLinhas;
t_cabecalho: Cab;
t_dados:    AreaDados[MAX_LINHAS];
t_rodape:   Rod;

- Montar cabeçalho do relatório
para DataInicial ate DataFinal faca
  - Validar a data tratada
  se (Data é válida)
    então
      - Fazer leitura do arquivo de supervisão da data tratada
      para Cubainicial ate CubaFinal faca
        - Montar área de dados da cuba para o dia tratado
        NLinhas = NLinhas + 1;
      fim para
    fim se
  fim para

- Montar rodapé do relatório

```

Copyright © 2012 Accenture. All rights reserved.

144

Criação de Relatórios

Procedimento de Criação de um Relatório

- No exemplo seguinte utilizaremos como exemplo o relatório diário **RelUserDia**, mas lembramos que o procedimento é o mesmo tanto para relatório de ciclo de controle quanto de turno
- 1º Passo** : implementar os arquivos fonte **RelExemploDia.h** e **RelExemploDia.c**
- 2º Passo** : criar variáveis e incluir cálculos no algoritmo de controle se necessário
- 3º Passo** : incluir as dependências no makefile para gerar os relatórios diários e turno e compilar


```
# cd /score/srcs
# GeraScore
```
- 4º Passo** : criar os descritores **RelExemploDia.dsc** (diário), **RelExemploTDia.dsc** (turno), etc ...

Copyright © 2012 Accenture. All rights reserved.

145

Criação de Relatórios

Procedimento de Criação de um Relatório

- 5º Passo** : criar os arquivos de help **RelExemploDia.hlp** (diário), **RelExemploTDia.hlp** (turno), etc ...
- 6º Passo** : configurar a chamada do programa na IHM pelo **ConfIHMPH** visto no capítulo *Ferramentas de Configuração*.
- 7º Passo** : depurar e testar o relatório
- 8º Passo** : preparar um upgrade com os arquivos alterados conforme capítulo *Preparação e Instalação de Upgrades*
 - Os relatórios são carregados em memória somente quando chamados via IHM. Portanto, se não houver tarefa residente incluída no upgrade não precisa desativar o sistema para a instalação.

Copyright © 2012 Accenture. All rights reserved.

146

Criação de Relatórios

Depuração de Relatórios

- Preparação do Programa para Depuração**
 - Incluir no makefile a diretiva **-g** (macro **DEBUG**)
 - Preparar o programa para depuração pelo **wd**
 - Incluir o comando **sleep(30)** logo no início do programa
 - Inibir os comandos de alocação e liberação de semáforos
 - Compilar e gerar os módulos executáveis do programa alterado com a diretiva **-g**
 - Depurar o programa no ambiente de desenvolvimento com o sistema de controle e interface gráfica rodando

Copyright © 2012 Accenture. All rights reserved.

147

Criação de Relatórios

Depuração de Relatórios

Depuração do Relatório em Execução

- Partir o sistema de controle e interface gráfica
- Chamar o programa pela tela de relatórios da interface gráfica
- Pegar seu *pid* do processo através do comando **sin** imediatamente à ativação. Considerando o nome do processo começado com **Rel**, o comando deve ser executado numa console livre conforme a sintaxe abaixo :

```
# sin -P Rel
```
- Pegar o programa com o depurador **wd** passando como parâmetro o *pid* **PidRel** conforme a sintaxe abaixo :

```
# wd PidRel
```

 - Esse comando **wd** deve ser executado num intervalo de tempo inferior a 30 segundos (**sleep**). Depois de passado esse intervalo o programa é liberado para depuração passo-a-passo dentro do **wd**.

Copyright © 2012 Accenture. All rights reserved.

148

Criação de Relatórios

Depuração de Relatórios

Depuração do Programa Terminado por Erro de Execução

- Utilitário **dumper**
 - Gera um arquivo de depuração do programa **RelUserDia** de path **/score/erros/RelUserDia.dmp**
 - Deve ter sido ativado anteriormente conforme a sintaxe abaixo :

```
# dumper &
```
- Quando programa **RelUserDia** morre por erro de execução seu arquivo de dump é acessado pelo depurador conforme a sintaxe abaixo

```
# wd -trap=pmd /score/erros/RelUserDia.dmp
```

 - O depurador entra direto no ponto que ocorreu o erro
 - Selecionando as opções **Show → Calls**, em alguns casos mostra a sequência de chamada de funções antes do erro, permitindo voltar à condição anterior e analisar as variáveis.

Copyright © 2012 Accenture. All rights reserved.

149

Criação de Relatórios

Laboratório

- Abrir o arquivo fonte do exemplo de relatório de ciclo de controle e analisar
 - Montagem do cabeçalho, área de dados e rodapé
 - Função de leitura de tabelas
- Abrir o arquivo fonte do exemplo de relatório de supervisão e analisar
 - Varredura da função de montagem dos dados
 - Localizar a macro de acesso às variáveis de supervisão e entender seu detalhamento no prólogo Macros.h
 - Abrir o makefile e identificar como é gerado o relatório diário e turno

Copyright © 2012 Accenture. All rights reserved.

150

Índice

Apresentação
Ferramentas de Desenvolvimento
Arquitetura do Sistema
Estruturas de Dados
Algoritmo de Controle
Criação de Opção de Operação
Criação de Relatórios
Preparação e Instalação de Upgrades
Projeto Final

Copyright © 2012 Accenture. All rights reserved.

151

Preparação e Instalação de Upgrades

Resumo da Instalação das Alterações

- Procedimento de instalação de um upgrade:
 - Gerar o arquivo de upgrade e copiar para os micros que serão atualizados
 - Fazer backups do sistema como contingência antes de fazer a instalação
 - Instalar o upgrade num micro de controle, deixando o outro micro de controle e demais de supervisão desativados como contingência até comprovar que não há nenhum problema com a nova versão.
 - Instalar o upgrade no micro Hot-Standby e num micro de supervisão para validar a nova versão
 - Instalar no outro micro de controle e demais micros de supervisão depois que a nova versão estiver homologada.
 - Atualizar os backups da plataforma de desenvolvimento e runtime do sistema.
 - Atualizar o sistema das demais reduções seguindo o mesmo procedimento da 1ª redução instalada.

Copyright © 2012 Accenture. All rights reserved.

152

Preparação e Instalação de Upgrades

Procedimento de Preparação do Upgrade

- **1º Passo** : editar o arquivo `score/altcba/altcba.txt` e incluir versão, data e alterações implementadas conforme o padrão abaixo:

```
ALTERAÇÕES EFETUADAS NO SCORE REPORTS DO VERSAO DE 00-05-2012
VERSÃO 4.41
1) Alterado numero da versao de 4.40 para 4.41
2) Correcao da impressao de hardcopy em impressora colorida.
```

- **2º Passo** : preparar a lista de arquivos alterados
 - Gerar a lista pelo comando `filesd` conforme sintaxe abaixo:


```
# filesd AAAAMDDHHMM dir -v
```

```
# filesd 201209250000 /score/ | sort -k 8 >/tmp/lista
```

No exemplo são listados todos arquivos alterados a partir de 25/09/2012 às 00:00 ordenados pelo path e gravados no arquivo `/tmp/lista`
 - Editar o arquivo `/tmp/lista` e deixar somente a coluna com o path dos arquivos usando o comando `ALT+N`.

Copyright © 2012 Accenture. All rights reserved.

153

Preparação e Instalação de Upgrades

Procedimento de Preparação do Upgrade

- Limpar da lista os diretórios abaixo deixando somente o necessário:
 - **Base de Dados:** somente arquivos *MsgStr* e *Arqvar.dat*.
Os arquivos *Avc*, *cnfrelgraf*, *comum*, *evcuba*, *grupocubas*, *relgraf*, *superv_avc* e *valgrafmini* nunca podem ir no upgrade
 - **Hist, eventos, erros, exporta_dados, log, log_rel_cuba:** nunca vão
- Incluir no arquivo */tmp/lista* os arquivos *install_msg* (1ª linha), *altscore.txt* e *setup* se necessário (última linha)
- Copiar */tmp/lista* para o *path* */score/altcba/altrcbavxxxx_ddmmaaaa*, como por exemplo */score/altcba/altrcbav0015_25092012*
- **3º Passo** : preparar o arquivo de upgrade
 - Gerar o upgrade a partir da lista com mesmo path com a extensão *.tgz*, ou seja, */score/altcba/altrcbav0015_25092012.tgz*:

```
# cd /score/altcba
# upgrade altrcbavxxxx_ddmmaaaa altrcbavxxxx_ddmmaaaa.tgz
# upgrade altrcbav0015_25092012 altrcbav0015_25092012.tgz
```

Copyright © 2012 Accenture. All rights reserved.

154

Preparação e Instalação de Upgrades

Procedimento de Preparação do Upgrade

- **4º Passo** : copiar a lista e o arquivo de upgrade para o diretório */score/altcba* do nodo de controle e supervisão que será atualizado
 - **Observação:** um exemplo pode ser visto em */score/altcba*
- **5º Passo** : fazer os backups de segurança antes de começar a instalação a partir do micro de controle (Hot-Standby) que será instalado:
 - Base de dados
 - Runtime completo
 - Backup de runtime baseado na lista de arquivos que será instalada, caso precise retornar à versão anterior rapidamente:

```
# cd /score/altcba
# upgrade altrcbavxxxx_ddmmaaaa altrcbavxxxx_ddmmaaaa.ant.tgz
# upgrade altrcbav0015_25092012 altrcbav0015_25092012.tgz
```

Copyright © 2012 Accenture. All rights reserved.

155

Preparação e Instalação de Upgrades

Laboratório

- Forçar a compilação do sistema
- Preparar setup de instalação do descritor de eventos e *DescrArqVar.dat*
- Preparar um arquivo de upgrade com os seguintes arquivos:
 - Abertura com o *install_msg*
 - Arquivos compilados
 - Descritor de eventos e *DescrArqVar.dat*
 - Setup de instalação

Copyright © 2012 Accenture. All rights reserved.

156

Preparação e Instalação de Upgrades

Instalação do Upgrade no Hot-Standby

- **1º Passo** : desativar o micro Hot-Standby
 - Logo depois da atualização a cada 2 minutos no diretório `/score/base_dados`, o micro Hot-Standby deve ser desativado.
- **2º Passo** : instalar o upgrade já copiado para o diretório `/score/altcba` do HotStandby pelo comando ***install***:


```
# cd /
# install -u /score/altscore/altcbavxxx_dmmmaaa.tgz
# install -u /score/altcba/altcbav0015_25092012.tgz
```
- **3º Passo** : instalar nova tarefa residente pelo comando ***InstalaCtrl***
- **4º Passo** : desativar o micro que está controlando e partir o micro Hot-Standby com a nova versão para controlar a redução
- **5º Passo** : verificar se a nova versão do sistema está funcionando normalmente, movimentação de anodo, EA, corrida, etc...

Copyright © 2012 Accenture. All rights reserved.

157

Preparação e Instalação de Upgrades

Instalação do Upgrade no Hot-Standby

- **Observações** :
 - O Hot-Standby deve ser desativado para instalação somente quando o upgrade envolver **atualização ou inclusão de nova tarefa residente**. Não há necessidade de desativar o controle para instalar programas de operação e relatórios.
 - Na operação acima, os dados de base de dados, eventos, históricos e logs ocorridos no período entre a desativação e reativação do Hot-Standby serão perdidos. Uma forma de evitar que isso aconteça é reativar o Hot-Standby novamente como reserva e fazer a transferência depois de concluída a atualização dos dados. Nesse caso, se o upgrade envolver qualquer arquivo do diretório `/score/base_dados`, este deve ser atualizado manualmente.
 - Nenhum micro de supervisão deve acessar o micro de controle com a versão desatualizada.

Copyright © 2012 Accenture. All rights reserved.

158

Preparação e Instalação de Upgrades

Instalação do Upgrade nos Nodos de Supervisão

- **1º Passo** : desativar o micro de supervisão e abrir uma janela de comandos do shell do QNX.
- **2º Passo** : copiar o upgrade para seu diretório `/score/altcba`
- **3º Passo** : instalar o upgrade pelo comando ***install***:


```
# cd /
# install -u /score/altscore/altcbavxxx_dmmmaaa.tgz
# install -u /score/altcba/altcbav0015_25092012.tgz
```
- **4º Passo** : partir o sistema de supervisão depois que o controle já estiver rodando com a nova versão do sistema

Copyright © 2012 Accenture. All rights reserved.

159

Preparação e Instalação de Upgrades

Instalação do Upgrade nos Demais Nodos da Fábrica

- **1º Passo** : instalar o upgrade no restante dos nodos da redução testada seguindo os respectivos procedimentos dos micros de supervisão e controle
 - Instalar no outro micro de controle e partir como Hot-Standby
 - Instalar no restante dos micros de supervisão
- **2º Passo** : instalar o upgrade em cada uma das outras reduções seguindo o mesmo procedimento da 1ª redução:
 - Fazer backups da base de dados e runtime antes da instalação
 - Instalar no micro Hot-Standby e num micro de supervisão e acompanhar por um período
 - Instalar no restante dos micros de controle e supervisão
- **3º Passo** : atualizar os backups de todas reduções após atualização
 - Fazer backups da base de dados e runtime de cada redução
 - Fazer backup do micro de desenvolvimento

Copyright © 2012 Accenture. All rights reserved.

160

Preparação e Instalação de Upgrades

Comandos para Verificação dos Upgrades

- A integridade do arquivo de upgrade com extensão **.F** como **path_arq.F** pode ser verificados pelo comando **fcats**, conforme a sintaxe abaixo:


```
# fcats path_arq.F | pax [-v]
# fcats /score/altcba/altcbav0015_25092012.F -v
```
- A integridade do arquivo de upgrade com extensão **.tgz** como **path_arq.tgz** pode ser verificados pelo comando **zcat**, conforme a sintaxe abaixo:


```
# zcat path_arq.tgz | pax [-v]
# zcat /score/altcba/altcbav0015_25092012.F -v
```

Copyright © 2012 Accenture. All rights reserved.

161

Preparação e Instalação de Upgrades

Laboratório

- Ler o arquivo de upgrade preparado no laboratório anterior

Copyright © 2012 Accenture. All rights reserved.

162

Índice

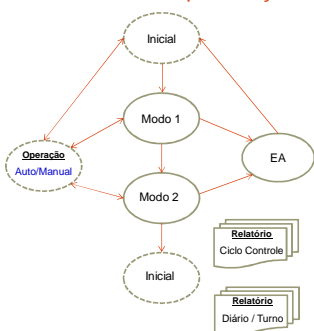
Apresentação
Ferramentas de Desenvolvimento
Arquitetura do Sistema
Estruturas de Dados
Algoritmo de Controle
Criação de Opção de Operação
Criação de Relatórios
Preparação e Instalação de Upgrades
Projeto Final

Copyright © 2012 Accenture. All rights reserved.

163

Projeto Final

Laboratório – Especificação Algoritmo de Alimentação



- Especificação Funcional:
 - **Startup Sistema**
 - Alimentação Inicial
 - **Operação Manual**
 - **Início:** Suspende Alimentação
 - **Fim:** Volta Alimentação Anterior
 - **Efeito Anódico**
 - **Início:** Alimentação EA
 - **Fim:** Volta Alimentação Inicial
 - **Modos de Operação:**
 - Inicial
 - Modo 1
 - Modo 2
 - EA
 - Suspensa

Copyright © 2012 Accenture. All rights reserved.

164

Projeto Final

Laboratório – Especificação Algoritmo de Alimentação

- Tarefa Residente MCP
 - Implementa mudança de estados
 - **Observação:** O MCP não pode aguardar retorno de comando da remota
- Tarefa Residente MCC
 - Implementa função para mudança de acionamento na remota
 - Geração de evento
- Programa de Operação
 - **Entrada:** Auto / Manual
 - **Ação:** Suspende Alimentação
 - Geração de evento

Copyright © 2012 Accenture. All rights reserved.

165

Projeto Final

Laboratório – Especificação Algoritmo de Alimentação

• Relatório de Ciclo de Controle

Relatório de Ciclo de Controle de Alimentação

-Inicial-	-Modo 1-	-Modo 2-	---E A---	-----Modo Atual-----
Cuba	Dur Kg	Dur Kg	Dur Kg	dd/mm/aa hh:mm Modo
xxx+	xx:xx xxx	xx:xx xxx	xx:xx xxx	xx/xx/xxxx xx:xx xxxxxxxx

Med

• Relatório Diário / Turno

Relatório Histórico de Alimentação – Turno x

-Inicial-	-Modo 1-	-Modo 2-	---E A---	-----Modo Atual-----
Cuba	Dur Kg	Dur Kg	Dur Kg	dd/mm/aa hh:mm Modo
xxx+	xx:xx xxx	xx:xx xxx	xx:xx xxx	xx/xx/xxxx xx:xx xxxxxxxx

Med

Copyright © 2012 Accenture. All rights reserved.

166

Projeto Final

Laboratório – Implementação Algoritmo de Alimentação

• 1ª Fase – Implementação das Tarefas Residentes

- Criação e configuração de parâmetros e variáveis de processo
- Implementação da tarefa Mcp
- Implementação da função da tarefa Mcc
- Configuração de eventos (DescrEv e Help)
- Depuração e instalação no controle

• 2ª Fase – Implementação do Programa de Operação

- Implementação do programa de operação
- Configuração da chamada do programa
- Depuração e instalação no controle

Copyright © 2012 Accenture. All rights reserved.

167

Projeto Final

Laboratório – Implementação Algoritmo de Alimentação

• 3ª Fase – Implementação Relatório de Ciclo de Controle

- Criação das variáveis e inclusão dos cálculos nas tarefas
- Implementação do relatório
- Configuração dos descritores
- Criação do Help do relatório
- Configuração da chamada do relatório
- Depuração e instalação no controle

• 4ª Fase – Implementação Relatório Diário / Turno

- Criação das variáveis e inclusão dos cálculos nas tarefas
- Implementação do relatório
- Configuração dos descritores
- Criação do Help do relatório
- Configuração da chamada do relatório
- Depuração e instalação no controle

Copyright © 2012 Accenture. All rights reserved.

168