



Capítulo 1

Revisão do Sistema



Prof. Aparecido V. de Freitas
Doutor em Engenharia da Computação pela EPUSP
avfreitas@uscs.edu.br





CAPÍTULO 1

Revisão de conceitos

PDM

OS/400

PROMPT

OBJETOS

ARQUIVOS

HELP



?

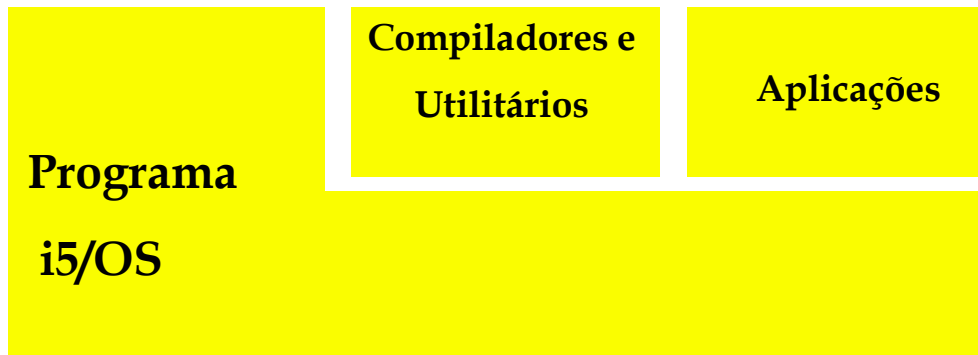
?

?



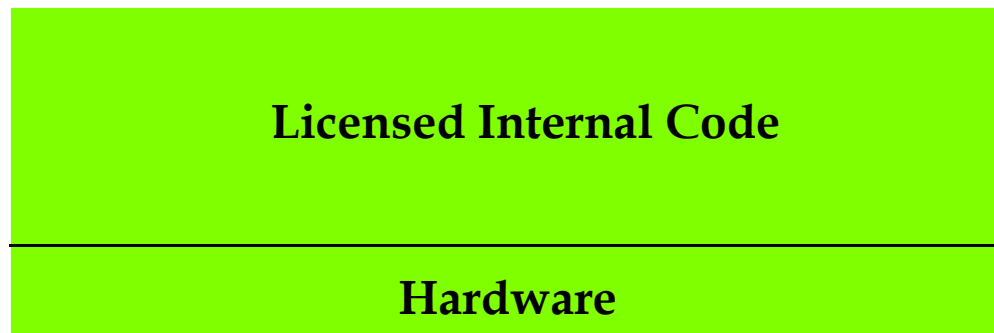


AS/400 – Arquitetura



Programa

High-Level Machine Interface
Instruction Set

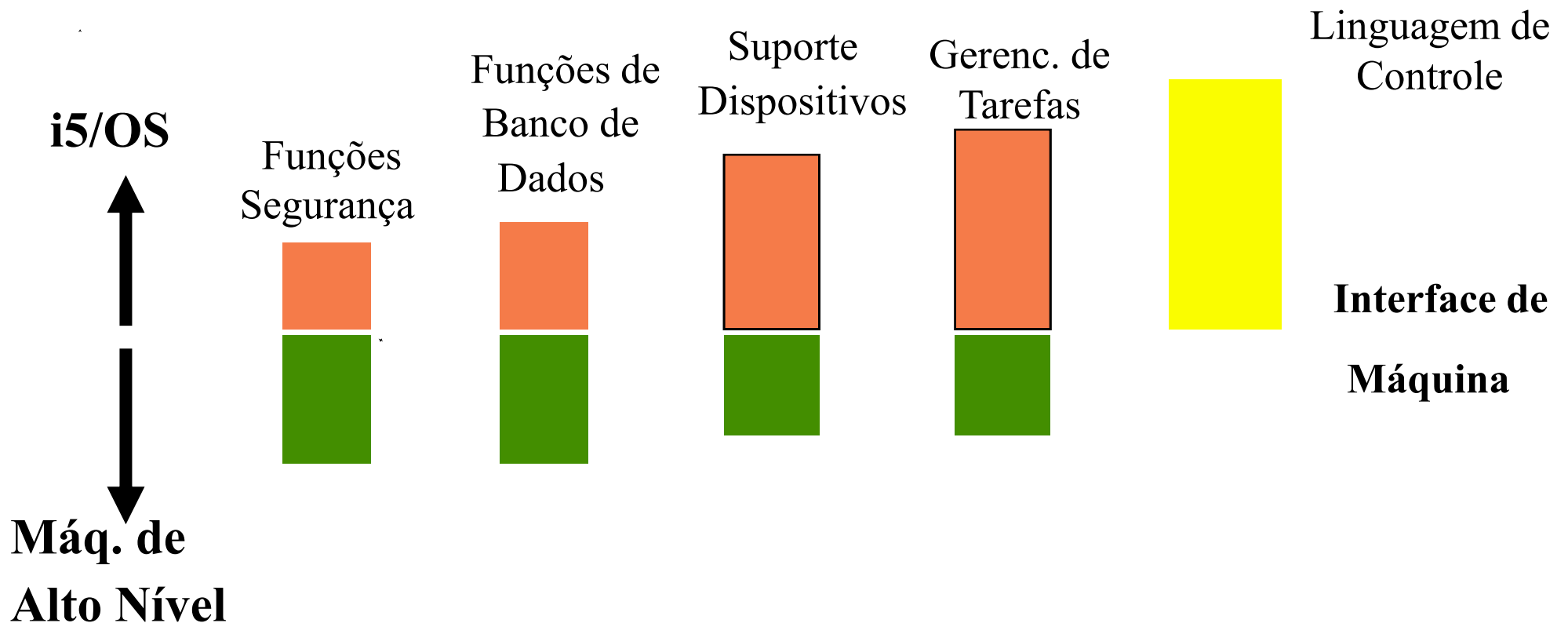


Máquina de
Alto Nível





Distribuição de Funções i5/OS e Máquina de Alto Nível





ARQUITETURA

- Máquina de Alto Nível
- Memória de Nível Único
- Orientação a Objetos





IBM i/OS

Permite a utilização de Menus e Comandos de controle CL(*Control Language*), para controlar e gerenciar dados.

Permite a execução de múltiplos JOBS interativos concorrentemente com JOBS BATCH, e o desenvolvimento de novas aplicações ao mesmo tempo.





Interface de Comandos

- Permitem a execução das funções do sistema
- Constituem a Linguagem de Controle (CL)
- Suporte de Prompting (F4)
- Suporte de Help e Index Search
- Podem ser recuperados (F9=Retrieve ou QCMD)





Comandos

Podem:

Executar programas	CALL PGM1
Copiar arquivos	CPYF ARQ1 ARQ2
Eliminar arquivos	DLTF ARQ2
Exibir Bibliotecas	DSPLIB *ALL
Desligar o sistema	PWRDWNSYS
Enviar mensagens	SNDBRKMSG
e muitas outras funções . . .	





Navegação de Menus

Main Menu

1. User tasks
2. Office tasks
3. System tasks
5. Programming

90. Signoff
Select or Command

==> _____

System Tasks

1. Jobs
2. Status
4. Messages

60. More system tasks

Select or Command

==> _____

Messages

1. Send a message
2. Send a break message
3. Display a message
4. Display a system...

70. Related commands
Select or Command

==> _____

F3=Exit

F16=AS/400 Main Menu

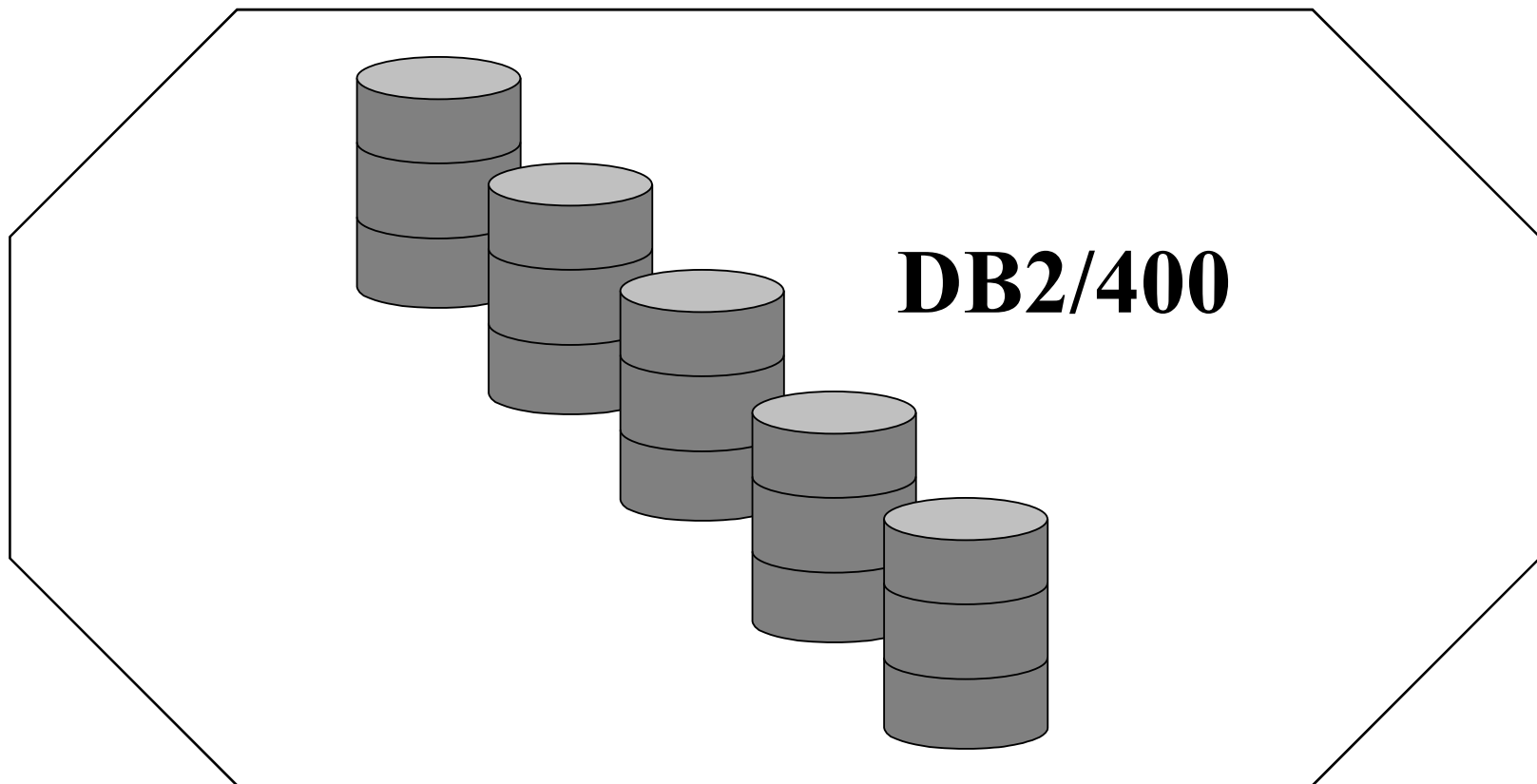
F12=Cancel

F23=Set initial menu





Banco de Dados





O modelo Relacional

- TABELA = ARQUIVO
- LINHA = REGISTRO
- COLUNA = CAMPO

No iSeries

TABELA = ARQUIVO FÍSICO

VISÃO = ARQUIVO LÓGICO





Arquivos de Tela (Display Files)

- Definem um formato de tela
- Formatam as informações e dados no terminal
 - » Constantes
 - » Campos de Entrada e Saída
 - » Manipulação de Indicadores
 - » Utilização de Teclas de Função
- Um arquivo de tela pode conter todos os formatos para um programa de aplicação





Arquivos de Impressão (Printer Files)

- Definem um formato de impressão
- Formatam as informações e dados no relatório
 - » Constantes
 - » Campos de Saida
- Um printer file pode conter todos os formatos para um relatório





ADT

Application Development Toolset





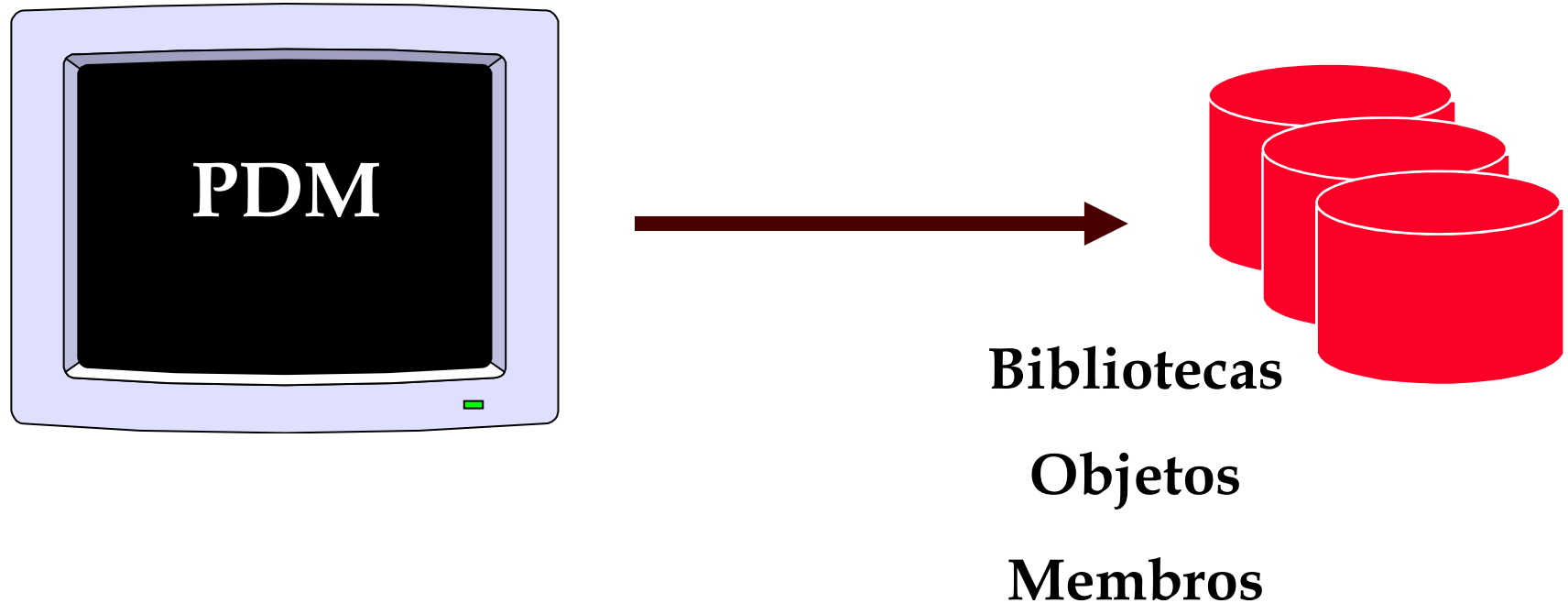
ADT – Application Development Toolset

- PDM Programming Development Tools
- SEU Source Entry Utility
- SDA Screen Design Aid
- DFU Data File Utility
- RLU Report Layout Utility



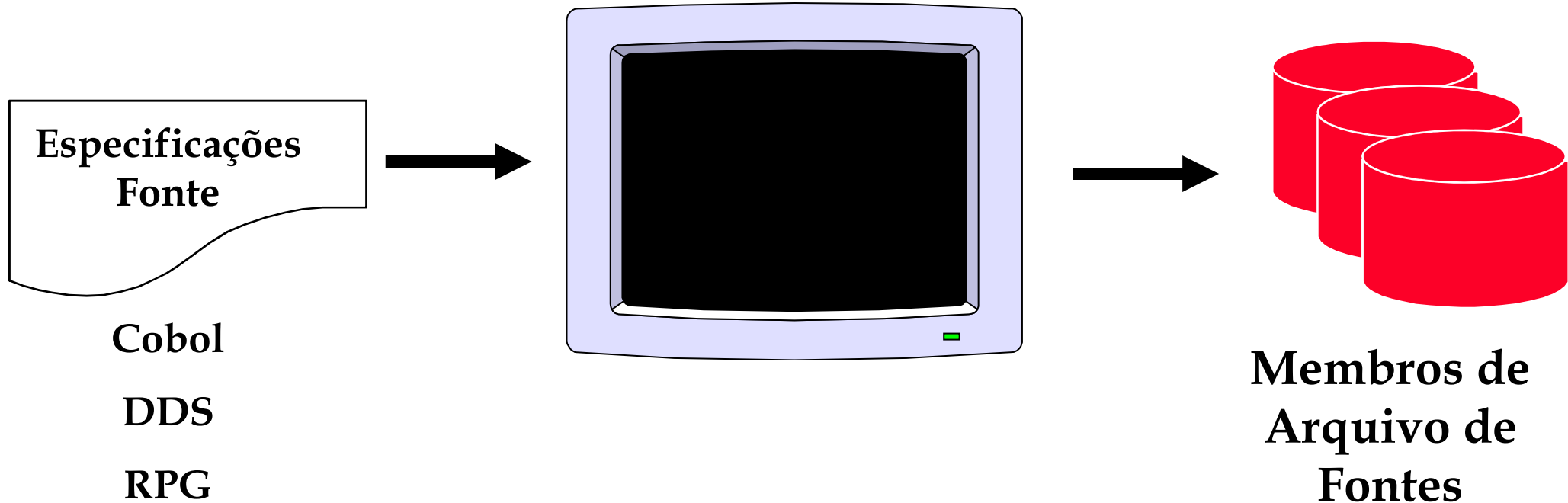


PDM - Programming Development Manager





SEU - Source Entry Utility





SDA - Screen Design Aid

- Projeto e/ou Alteração:
 - » Formatos de Registros de Telas
 - » Menus de Aplicações
 - » Subfiles
- Gera:
 - » Fontes DDS
 - » Display File
- Permite seleção de campos de Database





DFU

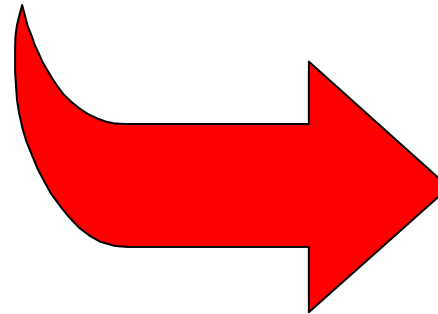
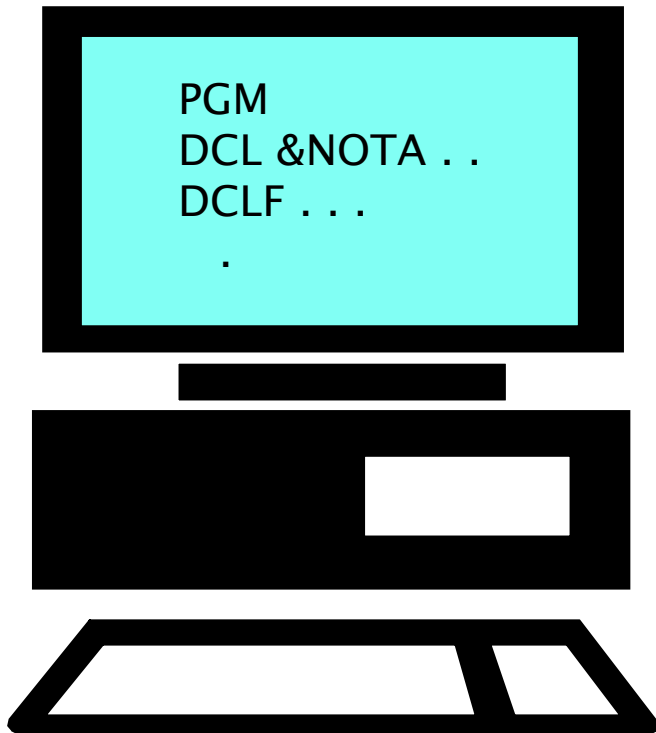
- Gera Programa Interativo
- Adição/Inclusão/Alteração de dados
- Consulta em arquivos de bancos de dados
- Obtenção rápida de resultados
- Não utiliza Linguagem de Programação





Capítulo 2

Programação CL Básica



Prof. Aparecido V. de Freitas
Doutor em Engenharia da Computação pela EPUSP
avfreitas@uscs.edu.br





Linguagem de Controle

- Linguagem do Sistema Operacional
- Controle de fluxo de execução de rotinas
- Operações Aritméticas e Lógicas
- Acesso a Bancos de Dados
- Acesso a Data Areas
- Compilação em código executável





Linguagem de Controle

- Interface com o Sistema Operacional
- Comandos representados por simples sentença
- Comandos podem ser entrados:
 - Individualmente em uma Workstation
 - Como sentença fonte para criação de programas CL
- Há prompting do i5/OS para todos os comandos
- Existem valores default para a maioria dos parâmetros





Linguagem de Controle

- 4Executar um programa CL é mais rápido que executar um comando CL único.
- 4Algumas funções só podem ser executadas em programas CL (*MONMSG, RTVSYVAL, IF, etc*).
- 4Programas CL podem ser testados e depurados antes de serem colocados em produção.
- 4Podem ser passados parâmetros





Linguagem de Controle

Funções

- 4Comunicação com o usuário.
- 4Acesso a Bancos de Dados
- 4Acesso a Data Areas
- 4Utilização de Display-Files
- 4Concatenação de variáveis
- 4Comandos de modificação de variáveis





Sintaxe de Comandos CL

[?] [Label-name:] [library-name/] command-name [parameter-set]

[]	Opcional
?	Prompt, como F4, para comandos fonte de programas CL. Precede label ou comando
label-name	Identifica comando para desvio, breakpoints e traces
library-name	Usualmente omitida, uma vez que comandos CL estão na QSYS





Regras de Codificação

- | | |
|--------------------|--------------------------------|
| • Nome da Variável | Deve iniciar com ‘&’ |
| • Dois Pontos | Separam labels de comandos |
| • Brancos | Separam comandos de parâmetros |
| • Parênteses | Separam Keywords e valores |
| • /* e */ | Comentários |
| • + ou - | Continuação na próxima linha |





PGM → Início do programa

Declarações
DCL
DCLF

Funções de Programa
SNDRCVF
CHGVAR
IF
ELSE

Passagem de Controle
CALL
SIGNOFF
RETURN
TRFCTL

ENDPGM → Término do programa





Estrutura de um programa CL

Início	PGM (opcional)
Declarações	DCLF (máximo de Um) DCL
Monitoração de mensagens ao nível de programa	MONMSG MONMSG
COMANDOS IBM	IF ELSE
E/OU comandos de Usuário	Monitoração de Mensagens ao Nível de campo
	COMMAND MONMSG MONMSG
Fim	ENDPGM





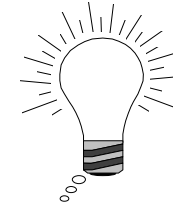
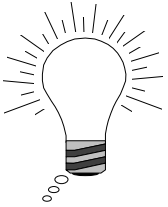
Declarando Variáveis

```
DCL  VAR(&TRAB) TYPE (*CHAR) LEN(10) VALUE('S026')  
DCLF FILE(IMESLIB/TESTEARQ)
```

- Devem começar com “&”
- O primeiro caractere deve ser alfabético
- Nome até 10 caracteres
- Devem ser declarados no início do programa
- Campos referenciados em uma DDS são automaticamente declarados no programa através do comando DCLF.



Declarando Variáveis



- Recomenda-se a utilização de PROMPT durante a definição de variáveis na linguagem de controle.
- Os tipos de variáveis podem ser *DEC, *CHAR, *LGL.
- Podem ser definidos valores iniciais às variáveis.

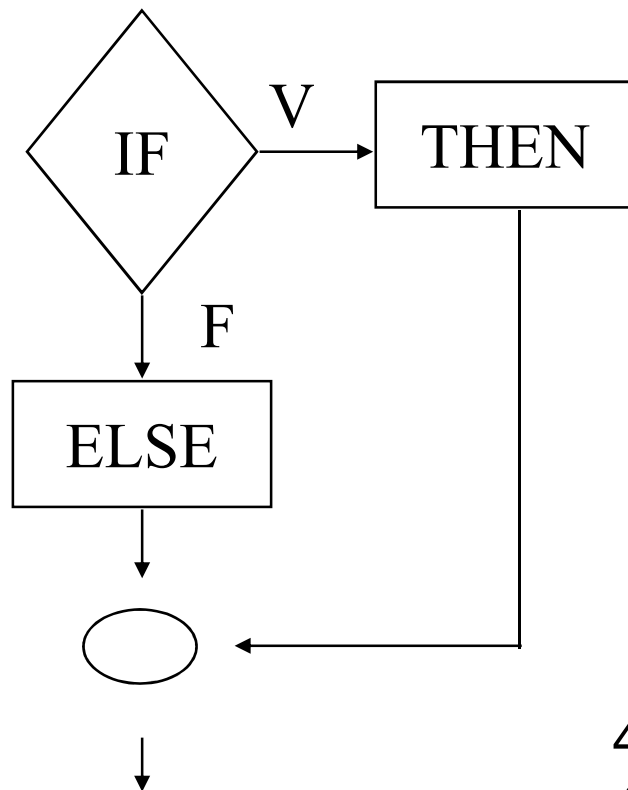


Declarando Variáveis

<i>TYPE ()</i>	<i>LEN ()</i>	<i>VALUE ()</i>
*DEC	Default: (15 5) Máximo: (15 9)	Default: (0)
*CHAR	Default: (32) Máximo: (9999)	Default: (b)
*LGL	1	Default: ('0')



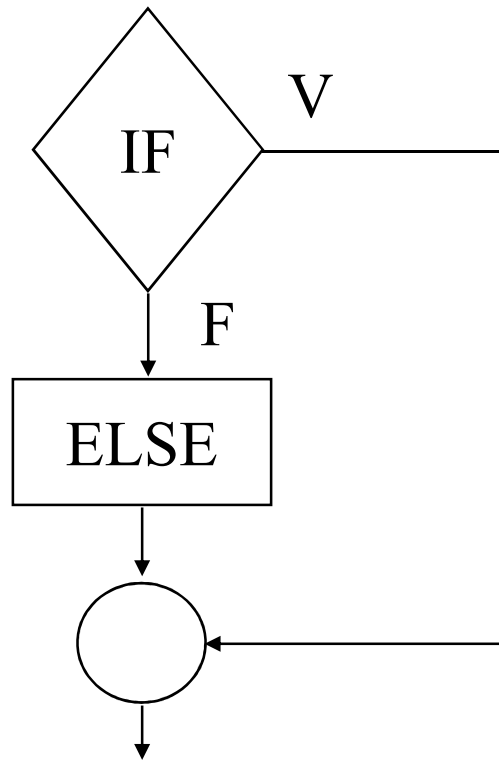
Processamento Condicional comando IF



relacional
IF **COND** or logical **THEN** (command)
expression
ELSE CMD (command)

4Os parâmetros THEN e ELSE são opcionais
4Apenas em programas CL

Processamento Condicional comando IF

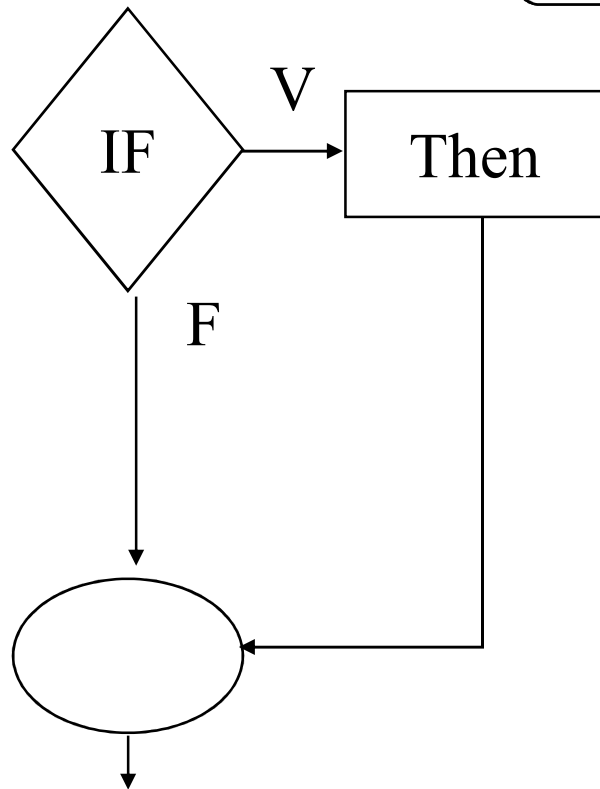


Cond. verdadeira: Controle de execução é seguido após o statment ELSE.

Cond. falsa: O comando no statment ELSE é executado.

4Parâmetro THEN omitido e ELSE especificado.

Processamento Condicional comando IF



- *Cond. verdadeira:* O comando no statment THEN é executado.
- *Cond. falsa:* Controle de execução é seguido após o statment IF.

4Parâmetro THEN especificado e ELSE omitido.



Expressões lógicas e Relacionais básicas

Operadores		Descrição
<	*LT	Menor que
=	*EQ	Igual a
>	*GT	Maior que
<=	*LE	Menor ou igual a
¬<	*NL	Não menor que
¬>	*NG	Não maior que
>=	*GE	Maior ou igual a
¬=	*NE	Diferente de
&	*AND	Verdade se ambos operandos são '1'
	*OR	Verdade se um dos operandos for '1'
¬	*NOT	Negação da condição





Exemplos de comandos IF

1. IF COND(&OPCAO *EQ 5) THEN (CALL PGM(INVENT))
2. IF (&QUANT > 0) THEN (CALL PGM1)
3. IF (&QUANTLIQ *GT 5000) SNDBRKMSG MSG('ERRO LIMITE')
ELSE CMD(GOTO ROTCALC)
4. IF (&A *NE &B) RETURN
5. IF (&X = &Y)
ELSE RETURN
6. IF (&IN01) GOTO ROTFIM





Comando DO

Delimita um bloco de comandos para serem processados.



Ex: IF COND(&X = &Y) THEN DO

•
•
•

ENDDO



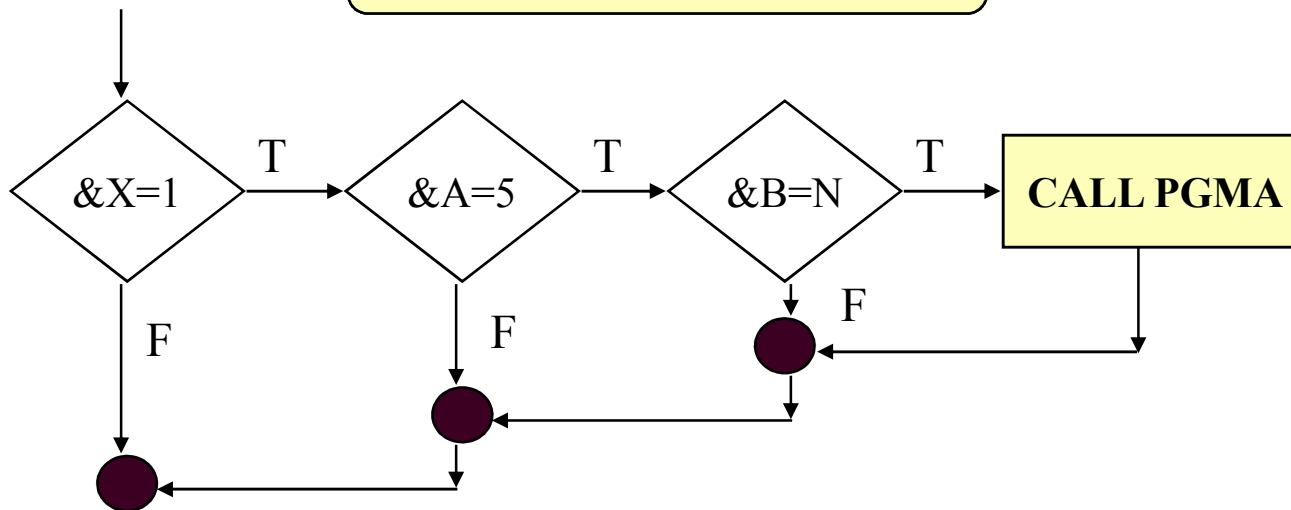


Comando DO

```
IF COND(&A *GT &B) THEN
    (DO)
        CALL PGM1
        CALL PGM2
    ENDDO
ELSE
    CMD(DO)
        CALL PGM3
        CALL PGM4
    ENDDO
    CALL PGM5
```



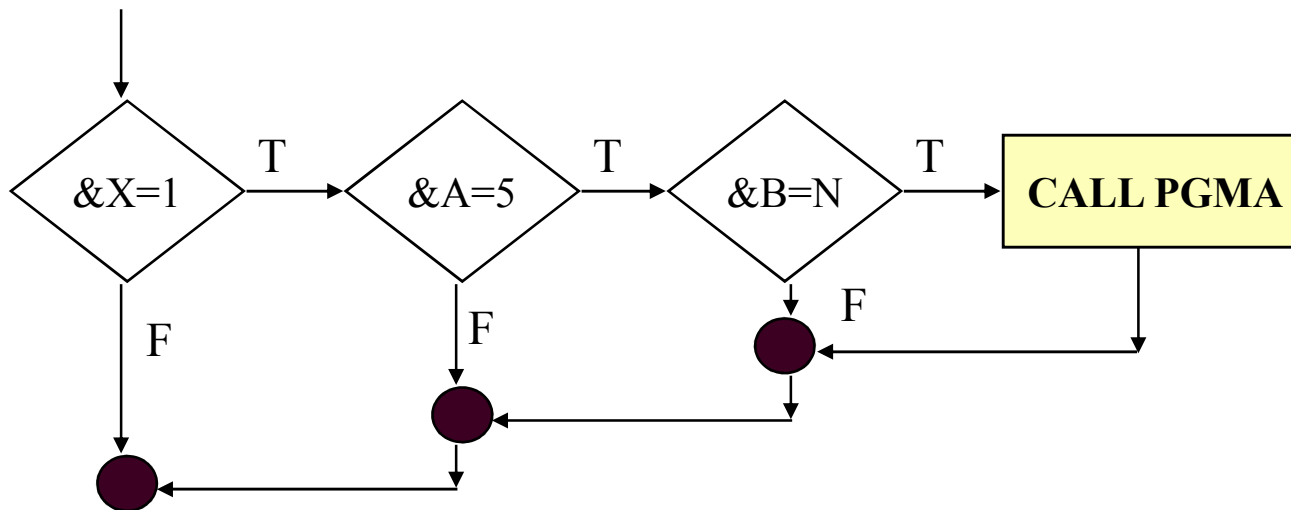
Aninhamento de IF's



Forma Keyword:

```
IF COND(&X=1) THEN +
  (IF COND(&A=5) THEN +
    (IF COND(&B=N) THEN +
      (CALL PGM(PGMA))))
```

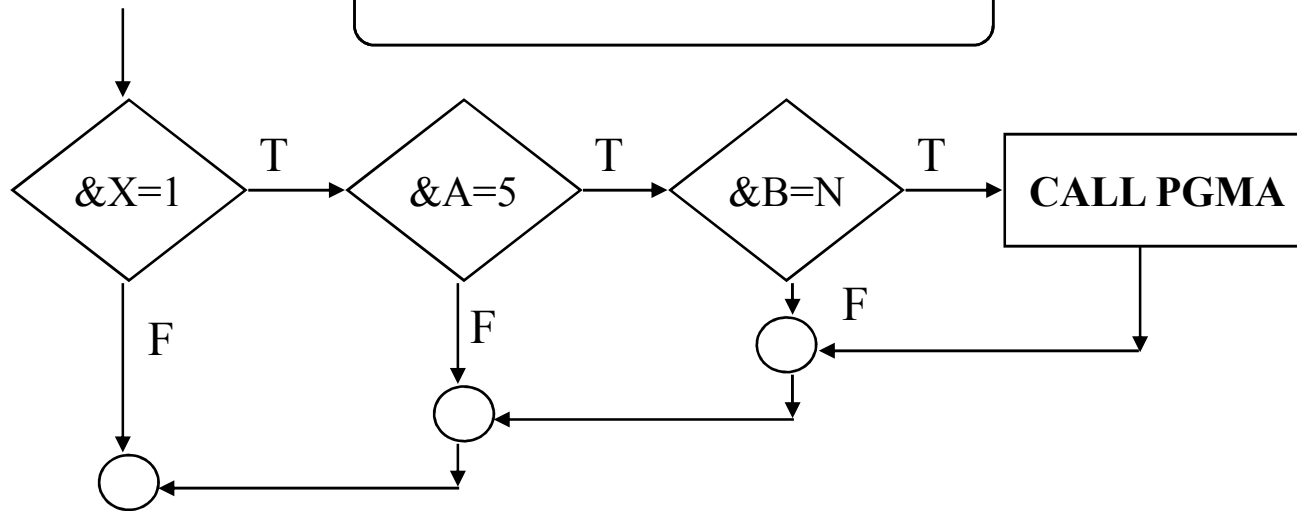
Aninhamento de IF's



Forma Posicional

IF (&X=1) IF (&A=5) IF (&B=N) CALL PGMA

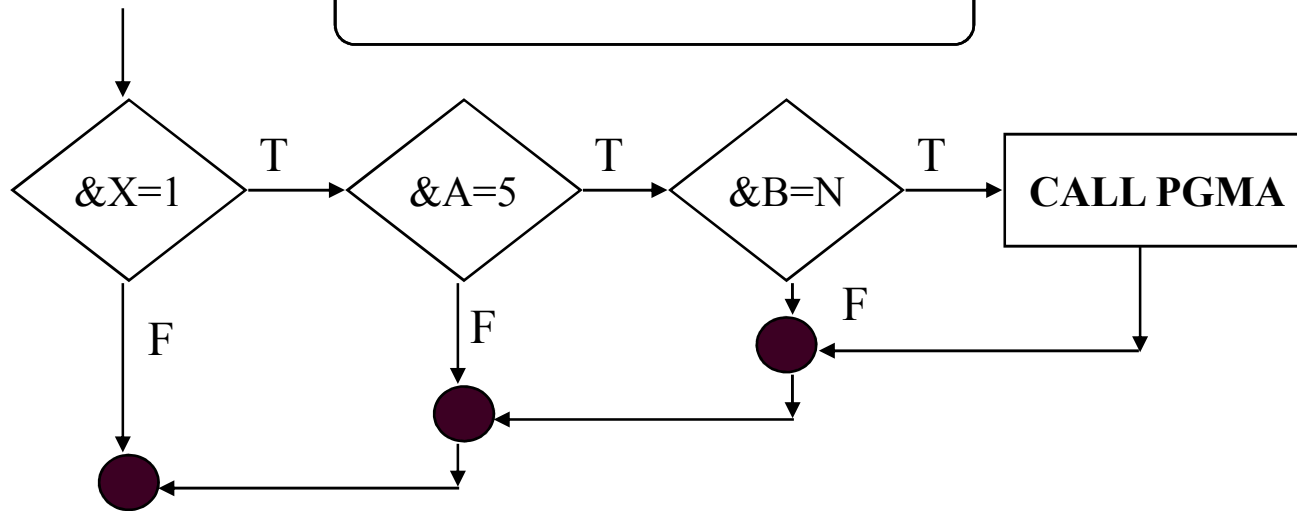
Aninhamento de IF's



Forma Agrupamentos DO, ENDDO:

```
IF COND(&X=1) THEN(DO) /* #1 */  
IF COND(&A=5) THEN(DO) /* #2 */  
IF COND(&B=N) THEN(DO) /* #3 */  
CALL PGMA  
ENDDO /* #3 */  
ENDDO /* #2 */  
ENDDO /* #1 */
```

Aninhamento de IF's



Um IF com condição composta:

```
IF COND(&X=1) *AND (&A=5) *AND +  
(&B=N) THEN(CALL PGMA)
```



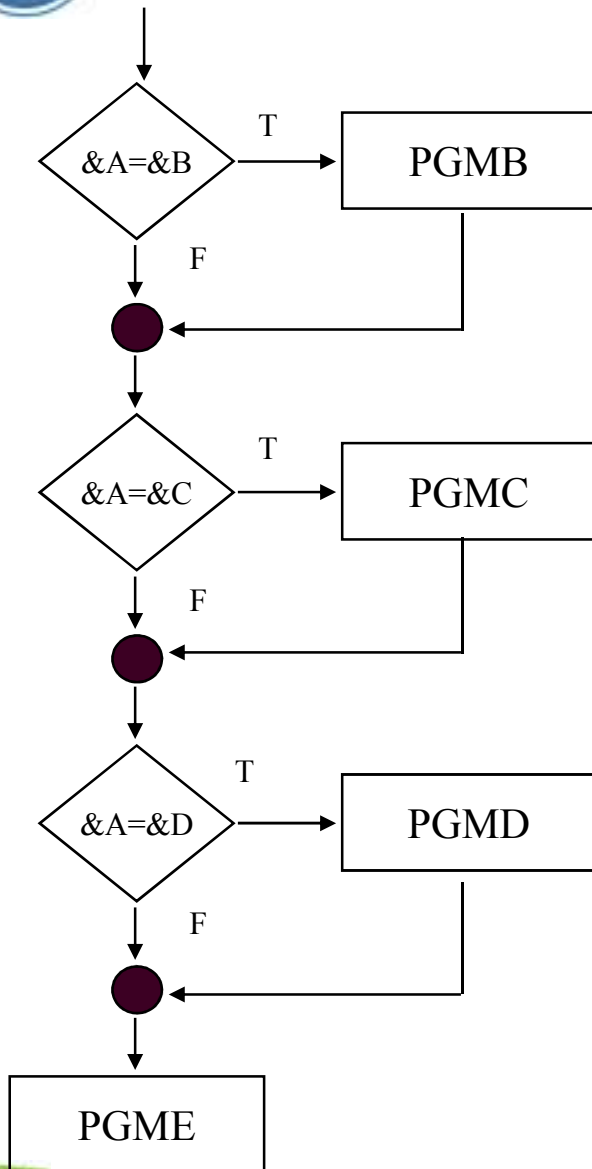
Séries de condições

PGMB E/OU PGMC E/OU PGMD

```
IF (&A=&B) CALL PGMB  
IF (&A=&C) CALL PGMC  
IF (&A=&D) CALL PGMD  
CALL PGME
```

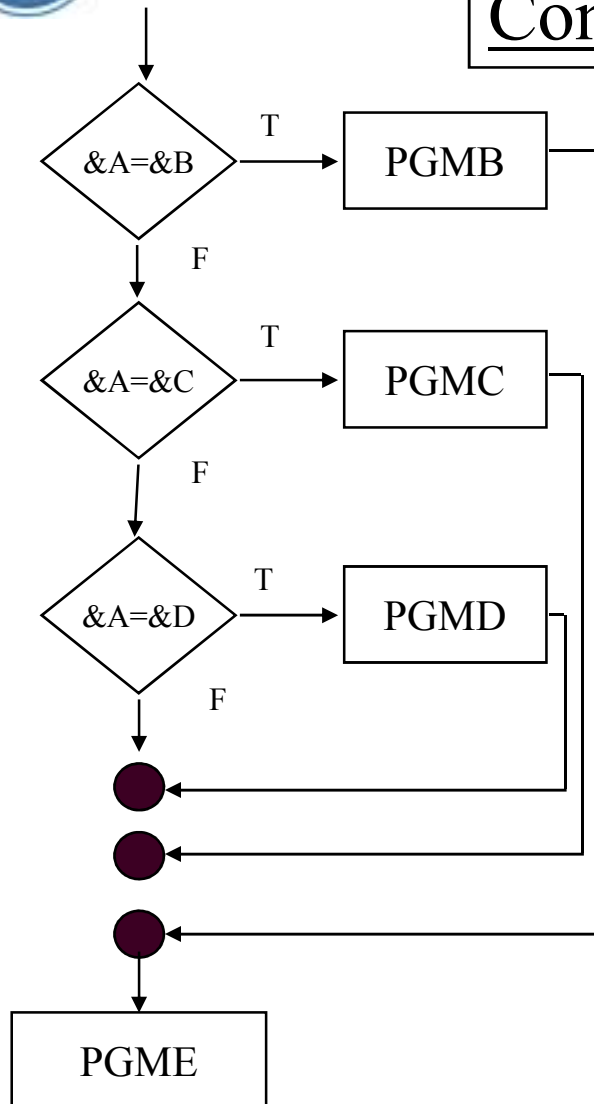
OU

```
IF COND(&A=&B) THEN (CALL(PGMB))  
IF COND(&A=&C) THEN (CALL(PGMC))  
IF COND(&A=&D) THEN (CALL(PGMD))  
CALL PGM(PGME)
```





Condições mutuamente exclusivas



PGMB OU PGMC OU PGMD

```
IF (&A=&B) CALL PGMB  
ELSE IF (&A=&C) CALL PGMC  
ELSE IF (&A=&D) CALL PGMD  
CALL PGME
```

OU

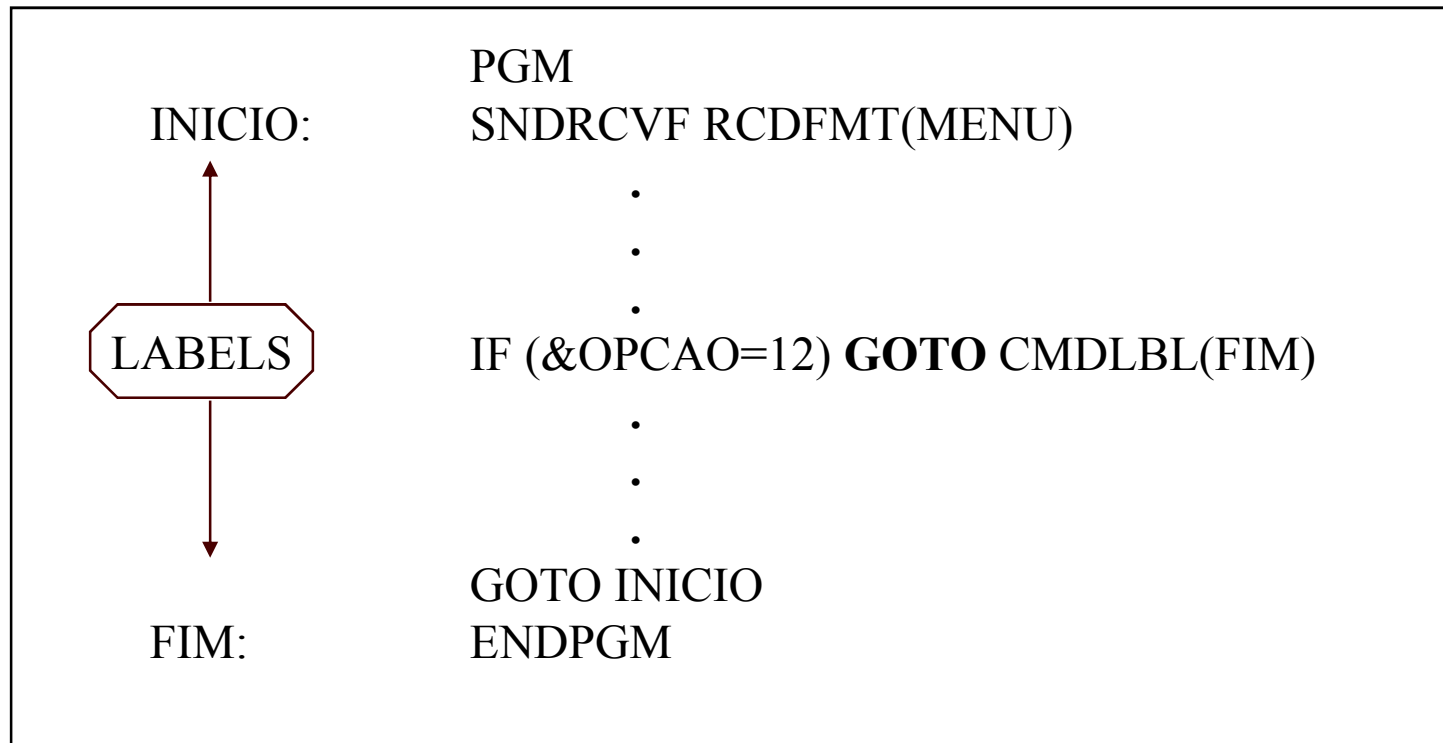
```
IF COND(&A=&B) THEN (CALL(PGMB))  
ELSE IF COND(&A=&C) THEN (CALL(PGMC))  
ELSE IF COND(&A=&D) THEN (CALL(PGMD))  
CALL PGM(PGME)
```





Comando GOTO

(desvio dentro do programa)



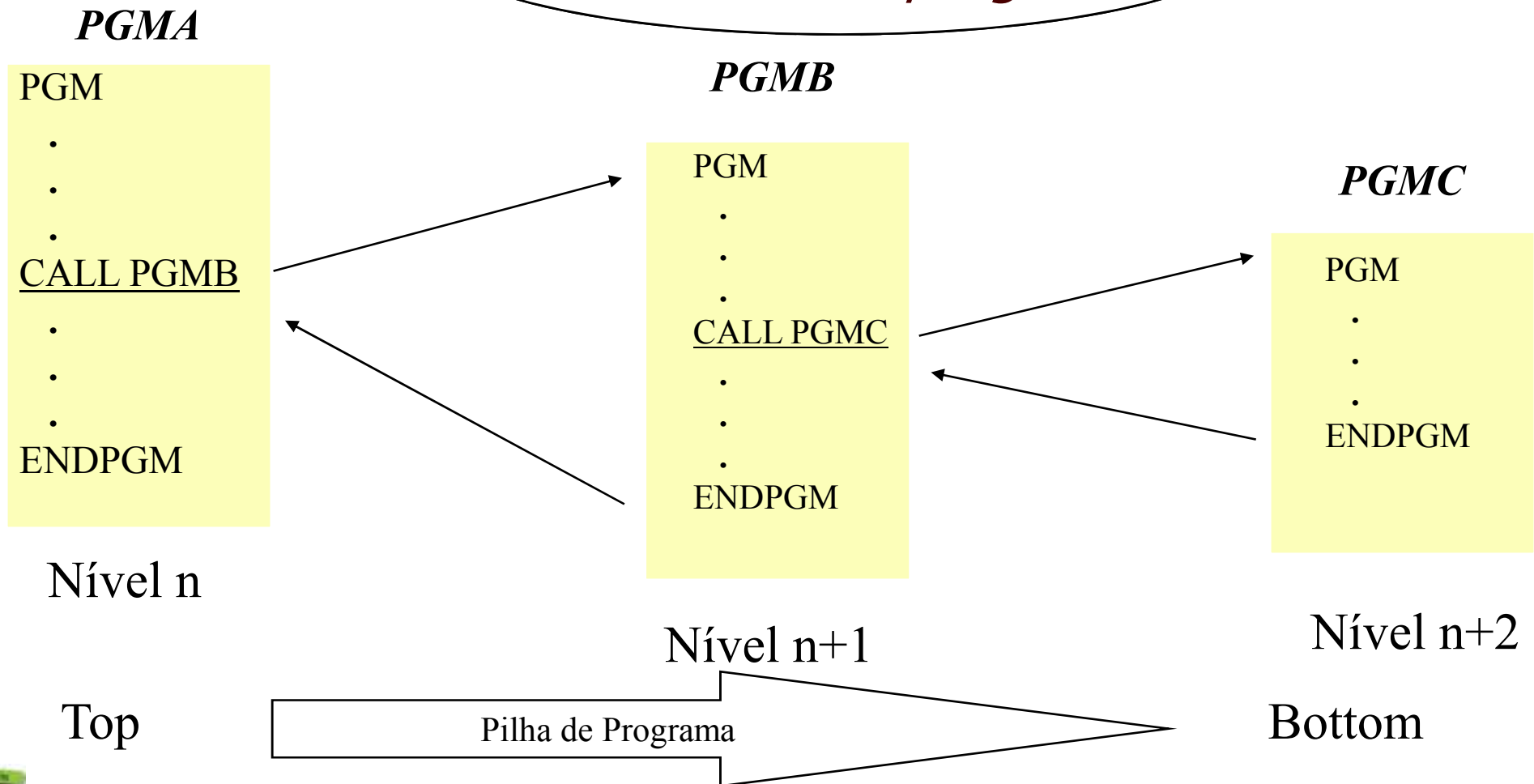
- r Só pode ser executado em programas CL
- r Processa uma ramificação/desvio incondicional
- r Labels devem ser únicos





Comando CALL

desvio fora do programa





Comando TFRCTL

desvio fora do programa

PGMA
Nível n

PGM
.
.
.
CALL PGMB
.
.
.
ENDPGM

PGMB
Nível n+1

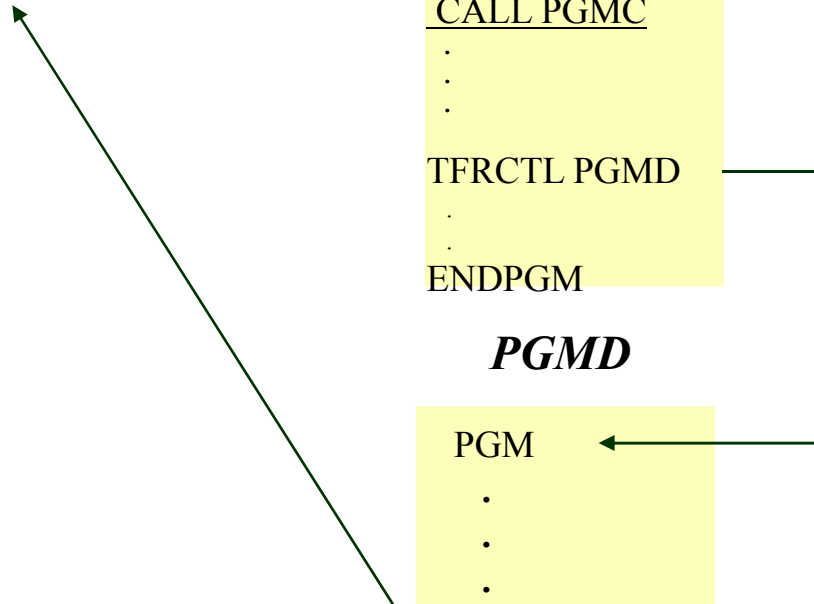
PGM
.
.
.
CALL PGMC
.
.
.
TFRCTL PGMD
.
ENDPGM

PGMC
Nível n+2

PGM
.
.
.
ENDPGM

PGMD

PGM
.
.
.
ENDPGM





Criação de um arquivo Display-File

*FILE DDSFONTES
Membro IMESMENU

R	IMESFMT		CA03(03 'EXIT')
		1 29	'Menu Principal'
		9 29	'Escolha a opção'
		4 29	'Consulta cadastro'
		5 29	'Atualização Ítems'
		6 29	'Cancela Ítems'
		7 29	'Sign off'
		11 29	'Opção'
OPCAO 2A I		11 39	
		13 29	'F3=FIM'

CRTDSPF



*FILE: IMESMENU

FORMAT: IMESFMT
Indicators: 03
Fields: OPCA0





Declarando um Display-File num programa CL

```
PGM  
DCLF FILE (IMESMENU)  
SNDRCVF RCDFMT(IMESFMT)  
.  
.  
.  
ENDPGM
```

3Máximo de 1 statement DCLF por programa.

3O programa não aloca memória para os campos com formatos não-referenciados.

3Campos dos formatos referenciados são implicitamente declarados no programa.

3Os programas CL não suportam campos decimais zonados
(convertidos automaticamente para decimais compactados).





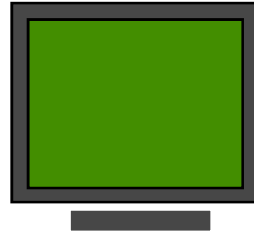
Declarando um Display-File num programa CL

```
PGM  PARM(&XYZ)
DCLF  FILE (IMESMENU)
DCL  VAR(&XYZ) TYPE (*CHAR) LEN (...)
DCL  VAR(&...)
      .
      .
      .
INICIO:
      SNDRCVF RCDFMT(IMESFMT)
      IF &IN03 RETURN
ELSE  IF COND (&OPCAO *EQ '1') THEN (CALL PGM1)
ELSE  IF COND (&OPCAO *EQ '2') THEN (CALL PGM2)
ELSE  IF COND (&OPCAO *EQ '3') THEN (CALL PGM3)
ELSE  IF COND (&OPCAO *EQ '4') THEN (CALL PGM4)
ELSE  IF COND (&OPCAO *EQ '90') THEN (SIGNOFF)
      GOTO INICIO
SEGUE:
      .
      .
      .
ENDPGM
```





Operações com Display File



DCLF	FILE(IMESMENU)
.	
.	
SNDF	RCFFMT(IMESFMT)
.	
.	
RCVF	RCDFMT(IMESFMT)
.	
.	
SNDRCVF	RCDFMT(IMESFMT) WAIT(*NO, *YES)
.	
.	
WAIT	





Passando parâmetro para um Programa CL



Programa CL PGM1

```
PGM PARM(&VAR1 &VAR2)
```

```
DCL VAR(&VAR1) TYPE(*CHAR) LEN(3)
```

```
DCL VAR(&VAR2) TYPE(*CHAR) LEN(4)
```

```
.
```

```
.
```

```
.
```

No prompt de comandos

```
CALL PGM(PGM1) PARM(ABC XYZW)
```





Passando parâmetro para um Programa CL

Programa CL PGM1

```
PGM  
  
DCL VAR(&REG) TYPE(*CHAR) LEN(3)  
.  
.  
.  
CALL PGM(PGM2) PARM(&REG 'XYZW')
```



Programa CL PGM2

```
PGM PARM(&VAR1 &VAR2)  
  
DCL VAR(&VAR1) TYPE(*CHAR) LEN(3)  
DCL VAR(&VAR2) TYPE(*CHAR) LEN(4)  
.  
.  
.
```





Passando dados para um Programa CL – *CALL*

PGM: A

PGM

DCL VAR(&COUNTA) TYPE(*DEC) LEN(3 0)

.

.

CALL PGM(B) PARM('0' &COUNTA 100 'ABC Co.')

.

.

.

PGM: B

PGM PARM(&IN99 &COUNTB &X &CO)

DCL &IN99 *LGL

DCL &COUNTB *DEC (3 0)

DCL &X *DEC (15 5)

DCL &CO *CHAR (20)

.

.

CHGVAR VAR(&COUNTB) VALUE(&COUNTB + 1)

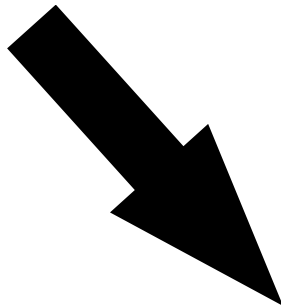




Chamadas de programas *HLL e CL*

*PGM: CLP

```
PGM
DCL VAR(&IN98)  TYPE(*LGL)
DCL VAR(&CUSTZ) TYPE(*DEC) LEN(5 0)
.
.
CALL PGM(COB1) PARM(&CUSTZ &IN98)
```



*PGM: COB1 - CBL

```
LINKAGE SECTION.
01      CUSTOMER  PIC S99999  COMP-3.
01      IN98      PIC 1.

PROCEDURE DIVISION
        USING CUSTOMER, IN98.
```





CHGVAR

Cálculos Aritméticos

CHGVAR VAR(CL-variável) VALUE(operando $\begin{matrix} + \\ - \\ / \\ * \end{matrix}$ operando)

CHGVAR VAR(&AMT) VALUE(-37.2)

CHGVAR VAR(&COUNT) VALUE(&COUNT + 1)

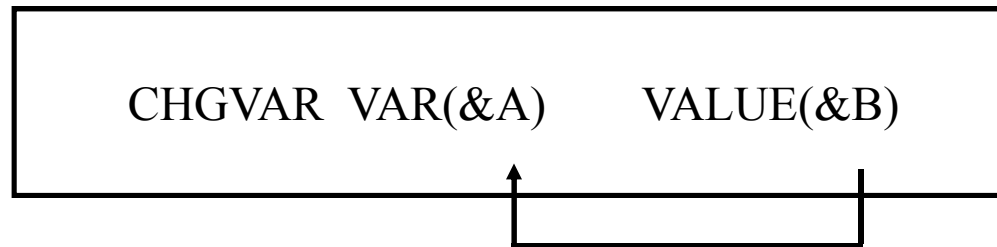
CHGVAR &AMT ((&PRECO - &DESCONTO) * &QTDE)





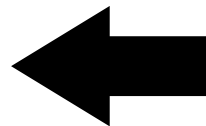
CHGVAR

Conversão entre tipos de dados



Variável Caractere &A

Length	Resultado Convertido
7	0023.00
7	-003.90
7	-123.67



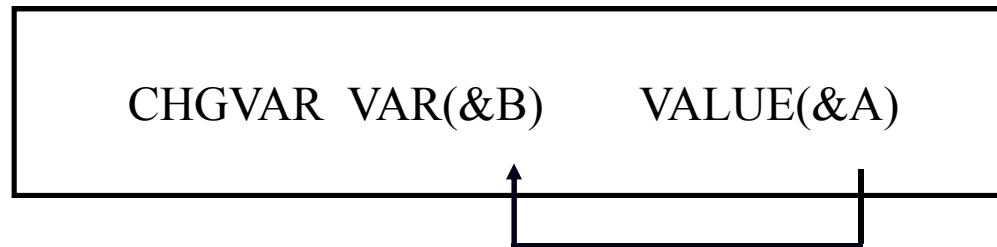
Variável Decimal &B

Length	Resultado Especificado
5,2	23.00 ou +23
5,2	-3.9
5,2	-123.67

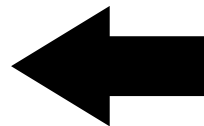




CHGVAR *conversão entre tipos de dados* **CHAR e *DEC*



Variável Decimal &B		Variável Caractere &A	
Length	Resultado Convertido	Length	Valor Especificado
5,2	123.10	10	'bb+123.1bb'
5,0	123	10	'bbb+123.00'
5,2	-123.00	10	'-123bbbbbb'





Concatenação de 2 variáveis ou valores do tipo *CHAR

*CAT - Concatenação sem Edição

&VAR1

JOSE

&VAR2

SILVA

*CAT

*CHGVAR VAR(&NOME) VALUE(&VAR1 *CAT &VAR2)*

&NOME

JOSE SILVA





Concatenação de 2 variáveis ou valores do tipo *CHAR

*BCAT - Brancos a direita do primeiro string são truncados.
Um branco é inserido. A concatenação é efetuada
e brancos a direita do segundo string não são truncados.

&VAR1

JOSE

&VAR2

*BCAT

SILVA

*CHGVAR VAR(&NOME) VALUE(&VAR1 *BCAT &VAR2)*

&NOME

JOSE SILVA





Concatenação de 2 variáveis ou valores do tipo *CHAR

*TCAT - Brancos a direita do primeiro string são truncados.
A concatenação é efetuada e brancos a direita do
segundo string não são truncados.

&VAR1

JOSE

&VAR2

SILVA

*TCAT

*CHGVAR VAR(&NOME) VALUE(&VAR1 *TCAT &VAR2)*

&NOME

JOSESILVA





Exemplo com concatenação de variáveis e conversão de tipos

```
PGM
DCL &NOMECLI          *CHAR  25      'IMES - COMPUTAÇÃO'
DCL &CODCLI           *DEC   (5 0)    54321
DCL &CODCLIALPH       *CHAR   5
DCL &DIAS             *DEC   (3 0)    30
DCL &DIASALPH         *CHAR   3

/* ALTERANDO O TIPO PARA CHAR */
CHGVAR &DIASALPH &DIAS
CHGVAR &CODCLIALPH &CODCLI
SNDMSG MSG('CLIENTE ' *BCAT &NOMECLI +
          *TCAT ', CONTA NÚMERO: ' *BCAT &CODCLIALPH +
          *CAT ', CREDITO LIBERADO POR ' *BCAT &DIASALPH +
          *BCAT 'DIAS.') TOMSGQ(MSGQ1)

ENDPGM
```

CLIENTE IMES - COMPUTAÇÃO, CONTA NUMERO: 54321,
CREDITO LIBERADO POR 030 DIAS.





Capítulo 3

Criação de programas CL





Listagem de compilação CL

- *SOURCE:** O compilador cria uma listagem do fonte de entrada usado para compilar o programa.
- *NOSOURCE:** A lista completa do programa fonte não é criada, apenas os erros de compilação são listados.
- *XREF:** O compilador cria uma lista de referência cruzada das definições de variáveis e labels no fonte.
Se *NOSOURCE é especificado, *NOXREF é sempre assumido.
- *NOXREF:** A lista de referências cruzadas não é gerada pelo compilador.





Listagem de compilação CL

- *GEN:** O compilador cria o programa objeto e o localiza na biblioteca especificada.
- *NOGEN:** Não é gerado o programa. O compilador efetua, checagem de sintaxe do programa fonte, e se *SOURCE for especificado é gerada a listagem fonte..
- *SECLVL:** Textos com nível de mensagens secundários serão impressos juntamente com os primeiros níveis de mensagens.
- *NOSECLVL:** Não serão impressos níveis secundários de mensagens.





Listagem de compilação CL

MESSAGE SUMMARY

Total	0-9	10-19	20-29	30-39	40-49	50-59	60-69	70-79	80-89	90-99
3	1	0	0	2	0	0	0	0	0	0

Program IMESPGM1 not created in library IMESLIB. Maximum severity 30.

***** END OF MESSAGE SUMMARY *****

***** END OF COMPILATION *****





Listagem de compilação CL

MESSAGE SEVERITY CODES

Severity	Descrição
00	Information
10	Warning
20	Error
30	Severe Error
40	Severe Error - Should not proceed
50	Abnormal termination of job or program
60	System Status
70	Device Integrity
80	System Alert
90	System Integrity
99	Action



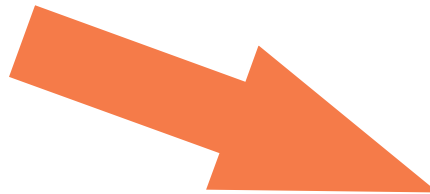


Recuperação do fonte de um programa CL – RTVCLSRC

OBJETO

Programa CL

IMESCL1



CLFONTES

IMESCL1

```
RTVCLSRC PGM(IMESCL1) SRCFILE(CLFONTES) SRCMBR(*PGM)
```





Recuperação do fonte de um programa CL – RTVCLSRC

0Comentários do programa fonte original não são recuperados.

0Os comandos são recuperados no formato Keyword.

0Um prólogo é adicionado ao novo membro fonte, especificando:

- Nome do proprietário (Owner)
- Data de última modificação do fonte original
- Data e hora do comando RTVCLSRC
- Nível do programa produto em tempo da compilação original.





Requisitando um DUMP do Programa CL

```
100   PGM
      .
      .
1700  ELSE  IF(&TYPE=S) CHGVAR &GROSS &SALARY
1800  ELSE  RETURN
1900  DUMPCLPGM
      .
      .
2100          ENDPGM
```

Planejado →





DUMP – Debugging Tool

5738 VnRnM 920424

CL Program Dump

Job Name: DSP02

User Name: José

Job Number: 014960

Program name: IMESPGM

Library: IMESLIB

Statment: 1900

Messages

Time	Message Id	Sev	Type	Message
<i>(No messages found on program's message queue)</i>				

Variable	Type	Len	Variables	Value in Hexadecimal
&EMPNO	*DEC	5 0	12345	
&GROSS	*DEC	6 2	215	
&HOUR	*DEC	3 1	40	
&OT	*DEC	6 2	15	
&OTHERS	*DEC	3 1	2	
&TYPE	*CHAR	1	'A'	C1





DUMP – Não Planejado

Display Formatted Message Text

System:IMES

Message ID: CPA0701
Message File: QCPFMSG
Library: QSYS

Possible Choices for replying to message

- C - Cancel the CL program
- D - Dump the CL program variables and cancel the CL program.**
- I - Ignore the failing command
- R - Retry the failing command again.

Não planejado





Display Program Reference DSPPGMREF

09/22/90 Display Program References

DSPPGMREF Command Input

Program: IMESPGM
Library: IMESLIB
Output: *
Text Description:
Number of objects referenced: 2
Object: PAYMAST
Library: *LIBL
Object Type: *FILE
Object: TAXPGM
Library: *LIBL
Object Type: *PGM

More...

F3=Exit F12=Cancel F19=Left F20=Right F24=More Keys





Lista de utilização de comandos PRTCMDUSG

5738SS1 VnRnMn **Print Command Usage**

Commands requested: *LIBL/CHGVAR

Programs requested: IMESLIB/*ALL

Command Use Detail

Program	Library	Command	Library
PGM1	IMESLIB	CHGVAR	*LIBL
PGM2	IMESLIB	CHGVAR	*LIBL

Libraries Searched: IMESLIB

CL Programs using commands: 35

CL Programs Searched: 60

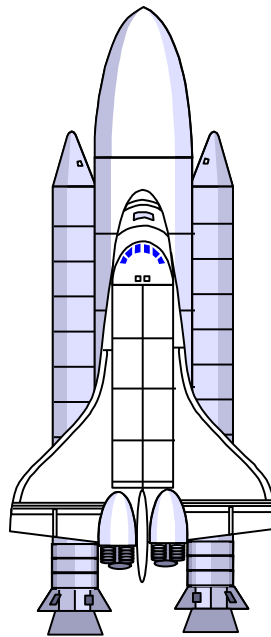
***** END OF COMMAND DETAIL *****





Capítulo 4

Programação CL Intermediária





Programação CL intermediária

- Funções %SUBSTR e %SWITCH
- Comandos de conversão de datas
- Recuperação e Modificação de atributos externos
- Data Areas
- Prompting para parâmetros em tempo de execução

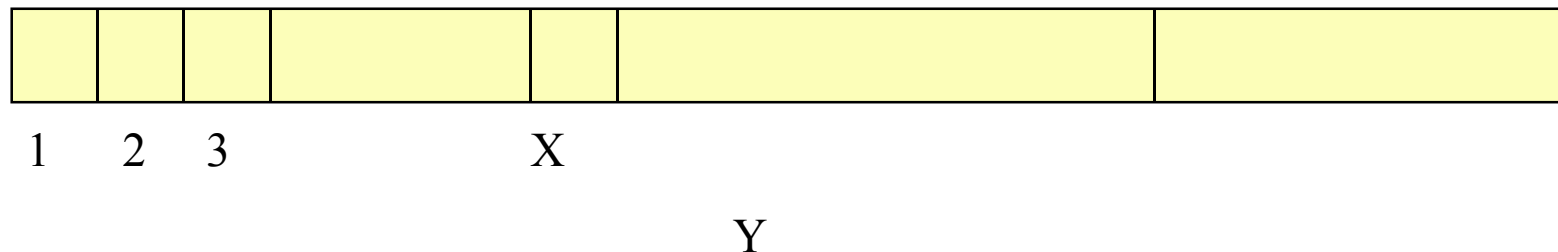




Função Substring (%SST)

- 3 Produz uma sequência de caracteres
- 3 Subconjunto de uma sequência de caracteres existentes
- 3 Usado apenas em programas CL

%SST (nome-de-variável X Y)



X = Posição inicial. Uma variável pode ser usada.

Y = Tamanho. Uma variável pode ser usada.





Exemplo do comando Substring

Exemplo: Extrair o número do time de seu User Name.

&USER

C	L	P	0	0	7				
---	---	---	---	---	---	--	--	--	--

CHGVAR VAR(&TEAM) VALUE(%SST(&USER 4 3))



&TEAM

0	0	7
---	---	---





Usando Substring para modificar uma variável

- *Para modificar o valor de uma variável pelo resultado de função %SST:*

```
CHGVAR VAR(&X) VALUE(%SST &A 1 5))
```

A variável &X assume o valor dos 5 primeiros caracteres do valor da variável &A

- *Para modificar parte de uma variável usando &SST:*

```
CHGVAR VAR(%SST(&A 4 3)) VALUE('XYZ')
```

O valor da variável &A é modificado em seu quarto, quinto e sexto caractere pela constante “XYZ”.





Exemplo – Função Substring

PGM

```
DCL VAR(&TEMP) TYPE(*CHAR) LEN(12) VALUE(ABCDEFGHIJKL)
DCL VAR(&LIB)   TYPE(*CHAR) LEN(6)   VALUE(IWLIBX)
DCL VAR(&X)     TYPE(*DEC)   LEN(2 0) VALUE(1)

INICIO: CHGVAR VAR(%SST(&LIB 6 1)) VALUE(%SST(&TEMP &X 1))

      SAVLIB LIB(&LIB)
      IF COND(&LIB='IWLIBL') THEN (GOTO FIM)
      CHGVAR VAR(&X) VALUE(&X + 1)
      GOTO INICIO

FIM:   ENDPGM
```

OBS: Este programa salva um grupo de bibliotecas com nomes de IWLIBA a IWLIBL.





Exemplo – Função Substring

PGM PARM(&STRG)

DCL	&WS	*CHAR	3
DCL	&X	*CHAR	1
DCL	&STRG	*CHAR	15

INICIO: CHGVAR &X %SST(&STRG 1 1)

CHGVAR &WS ('WS' *CAT &X)

IF (&WS='WS ') GOTO FIM

SNDMSG MSG('REUNIÃO ÀS 4:00hs') TOMSGQ(&WS)

CHGVAR &STRG %SST(&STRG 2 14)

GOTO INICIO

FIM: ENDPGM





Exemplo – Função Substring

```
CALL MSGPGM PARM('ABCDEFGHIJKLMNO')
```

- F Este programa envia uma mensagem para um grupo de workstations, cujos nomes variam somente pelo sufixo.
- F O programa efetua “Shifts Left” no string recebido como parâmetro.
- F As mensagens serão enviadas para as Messages Queues de WSA até WSN.





Local Data Area – *LDA

- 4 Área de 1024 bytes de dados caracteres criada e eliminada pelo Sistema em tempo de Signon.
- 4 Referenciada como *LDA
- 4 Restrita ao JOB
- 4 O comando SBMJOB passa a Local Data Area para o Job Batch

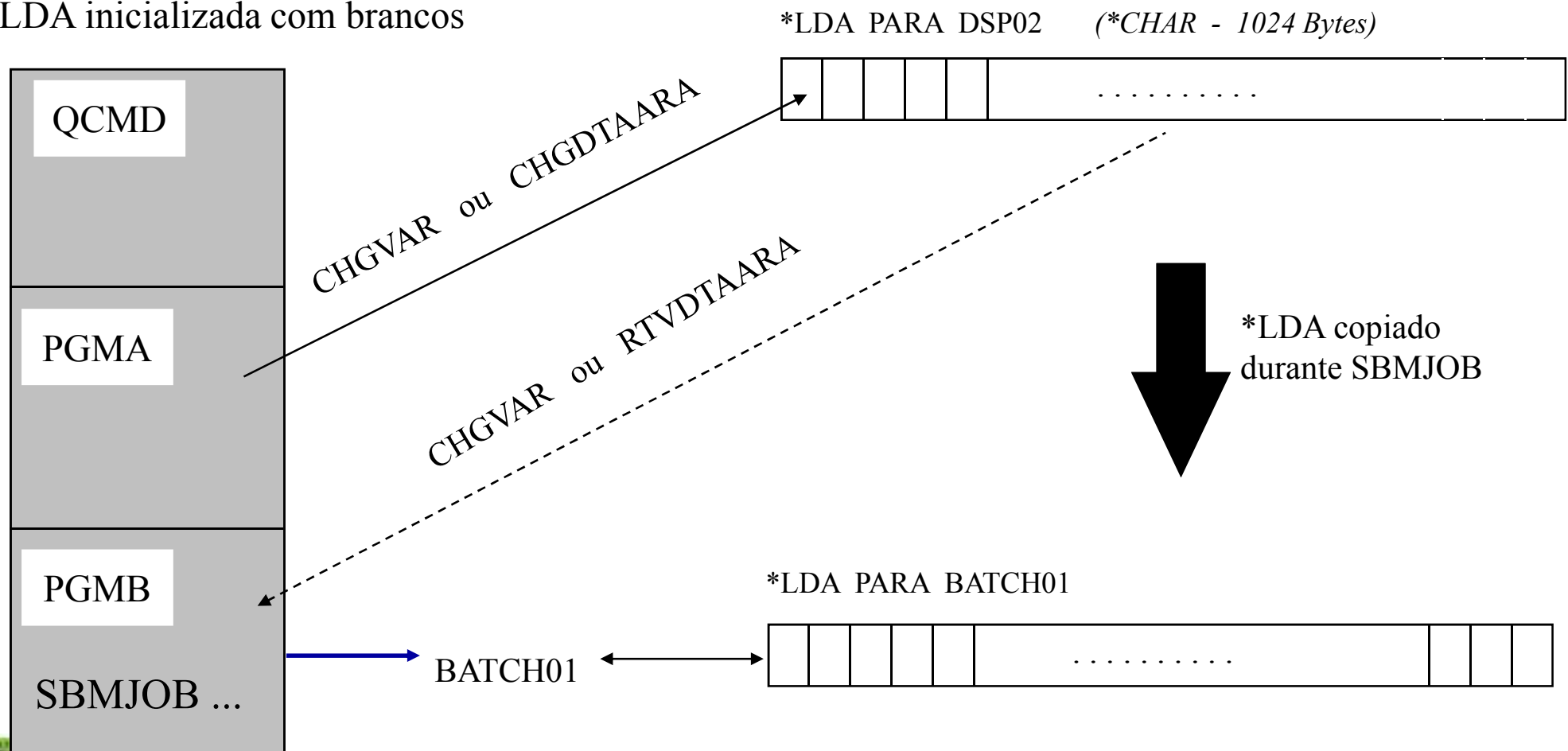




Local Data Area – *LDA

Signon: Início do Job DSP02

*LDA inicializada com brancos



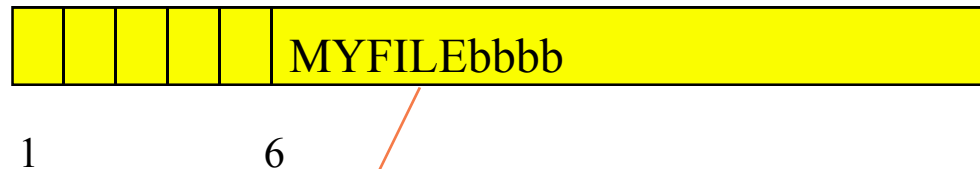
gnoff: *LDA eliminada



Acessando a Local Data Area

PROGA

```
PGM
CHGVAR  VAR(%SST(*LDA 6 10))  VALUE(MYFILE)
      ou
CHGDTAARA DTAARA(*LDA (6 10))  VALUE(MYFILE)
CALL PGMB
```



PROGB

```
PGM
DCL      VAR(&FILE)  TYPE(*CHAR) LEN(10)
CHGVAR  VAR(&FILE)  VALUE(%SST(*LDA 6 10))
      ou
RTVDTAARA DTAARA(*LDA (6 10))  RTNVAR(&FILE)
```





Acessando Local Data Area Cobol

ENVIRONMENT DIVISION.

SPECIAL-NAMES.

LOCAL-DATA IS LOCAL-DATA-AREA.

DATA DIVISION.

WORKING-STORAGE SECTION.

01 LDA-FIELDS.

05 A PIC X(4).

05 ORDNBR PIC X(5).

PROCEDURE DIVISION.

ACCEPT LDA-FIELDS FROM LOCAL-DATA-AREA.

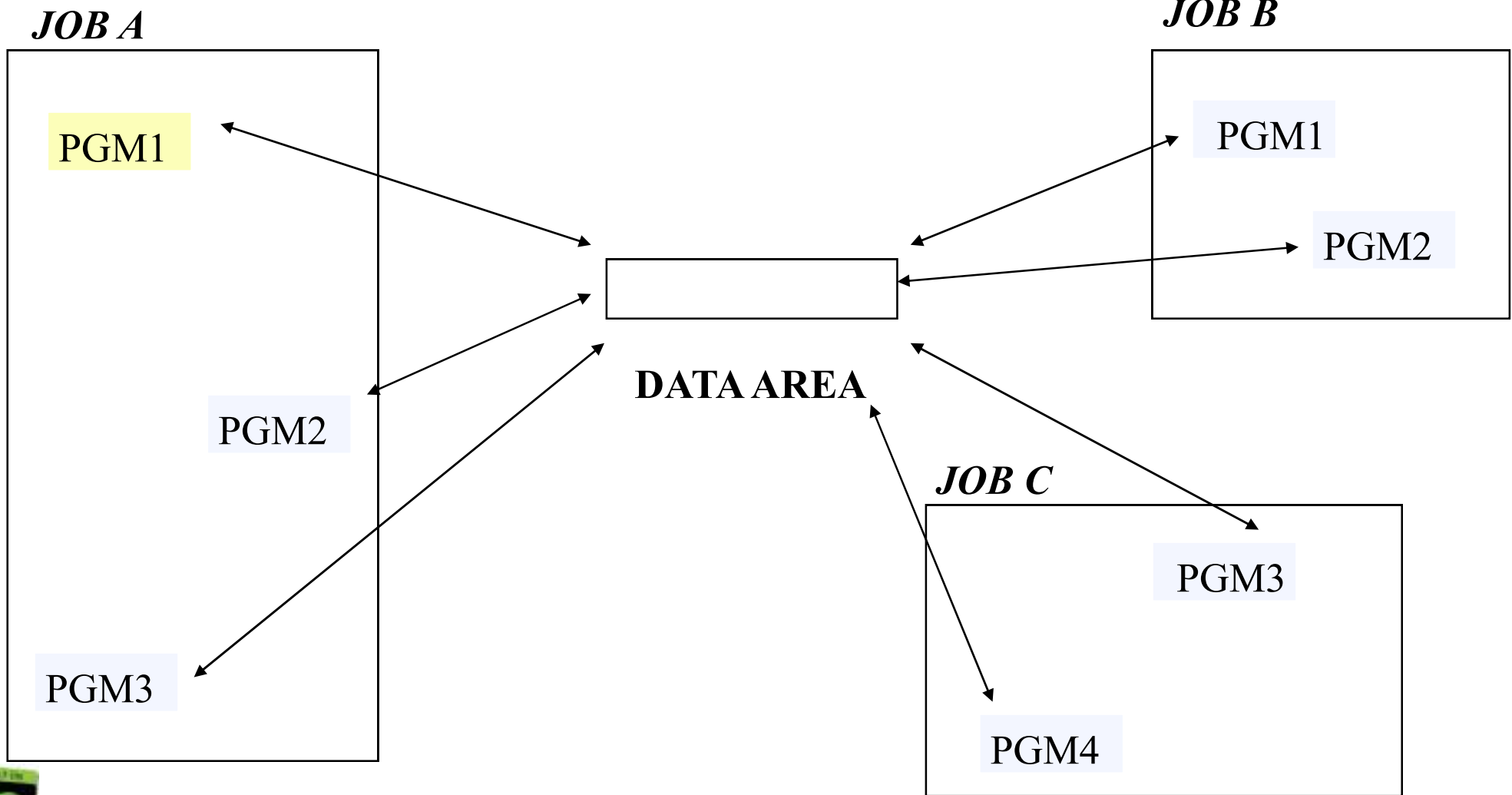
MOVE “12345” TO ORDNBR.

DISPLAY LDA-FIELDS UPON LOCAL-DATA-AREA.



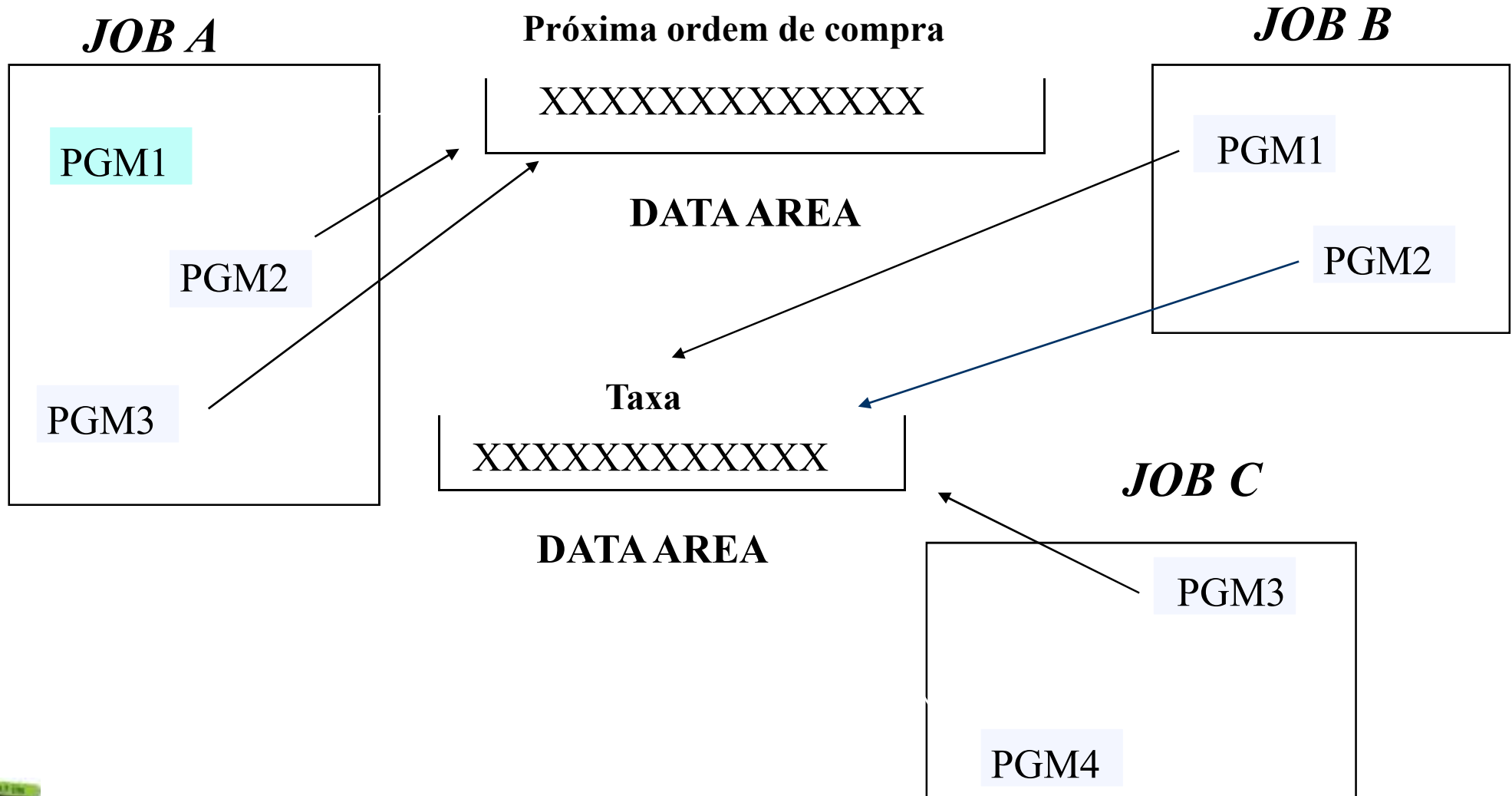


DATA AREA





Usos de Data Area





TIPOS DE DATA AREA

ì DATA AREA

ì LOCAL DATA AREA

ì GROUP DATA AREA

ì PIP DATA AREA

(Program Initialization Parameter)





TIPOS DE DATA AREA

Criado pelo

Tamanho

Tipo

Referenciado

Número p/ Job

Deletado pelo

<i>GENERAL DATA AREA</i>	<i>LOCAL DATA AREA</i>	<i>GROUP DATA AREA</i>	<i>PIP DATA AREA</i>
USUÁRIO	SISTEMA	SISTEMA	SISTEMA
Variável (1 - 2000 bytes)	Fixo (1024 bytes)	Fixo (512 bytes)	Fixo (2000 bytes)
*char, *dec, *lgl	*char	*char	*char
Interno ao Job ou entre Jobs	Interno ao Job	Interno à Group Job	Interno à Prestart Job
Variável	Um	Um	Um
USUÁRIO	SISTEMA	SISTEMA	SISTEMA





Comandos para manipulação de Data Areas

- **Create:** *CRTDTAARA*
- **Use:** *RTVDTAARA*
- **Return new value:** *CHGDTAARA*
- **Display:** *DSPDTAARA*
- **Delete:** *DLTDTAARA*





Comandos para Manipulação de Data Areas

Criação

CRTDTAARA

DTAARA(IMESLIB/ORDNBR) +
TYPE(*CHAR) LEN(5) +
TEXT('Próxima ordem de compra')

Uso num programa CL

RTVDTAARA

DTAARA(IMESLIB/ORDNBR (2 3) +
RTNVAR(&ORDER)

/* Cópia as posições 2, 3 e 4 de ORDNBR para a variável &ORDER */





Comandos para Manipulação de Data Areas

Armazenamento de um novo valor

CRTDTAARA DTAARA(IMESLIB/ORDNBR) +
VALUE(12345)

Display

DSPDTAARA DTAARA(IMESLIB/ORDNBR)

Delete

DLTDTAARA DTAARA(IMESLIB/ORDNBR)





Objetos Data Area (*DTAARA)

- *Próximo número sequencial - faturas, cheques, etc.*
- *Nomes - 'Instituto Municipal de Ensino Superior'*
- *Melhor que um *FILE para o armazenamento de pequenos volumes de informações.*



*CHAR

CRTDTAARA DTAARA(nome) TYPE(*DEC) LEN() VALUE()

*LGL

DSPDTAARA DTAARA(nome)

CHGDTAARA DTAARA(nome (início tamanho)) VALUE()

RTVDTAARA DTAARA(nome (início tamanho)) RTNVAR(variável CL)





Exemplo de programação utilizando Data Area

CRTDTAARA DTAARA(CHECKNO) TYPE(*DEC) LEN(5 0) VALUE(1)

```
PGM
DCLF  PAYOUT
DCL   &CHKFLD  *DEC(5 0)
```

```
SNDRCVF  RCDFMT(CHKENT)
```

```
ALCOBJ OBJ(CHECKNO *DTAARA *EXCL)
```

```
RTVDTAARA CHECKNO RTNVAR(&CHKFLD)
CHGVAR  &CHKFLD(&CHKFLD + 1)
CHGDTAARA CHECKNO(&CHKFLD)
```

```
DLCOBJ OBJ(CHECKNO *DTAARA *EXCL)
```

```
CALL CHECKWRITE(&NOME &AMT &CHKFLD)
ENDPGM
```

Entre com o nome
e quantidade:

Nome: _____
Qtd.: _____

***DTAARA: CHECKNO**

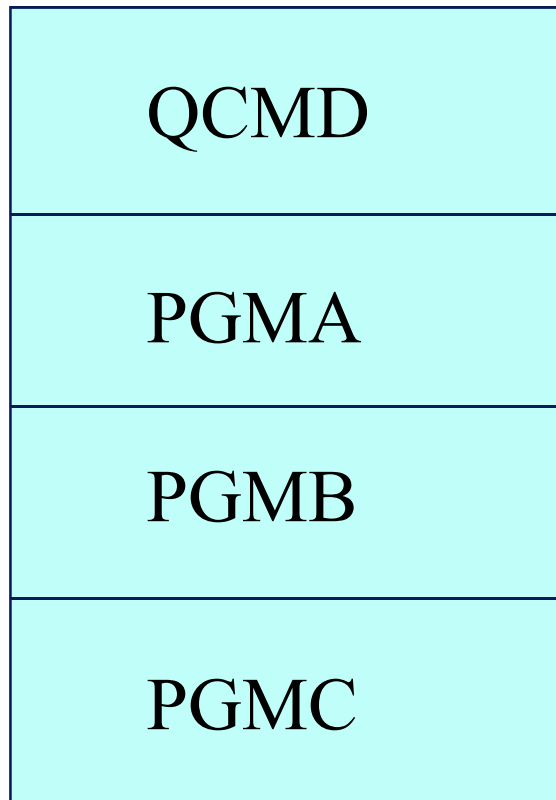
00001





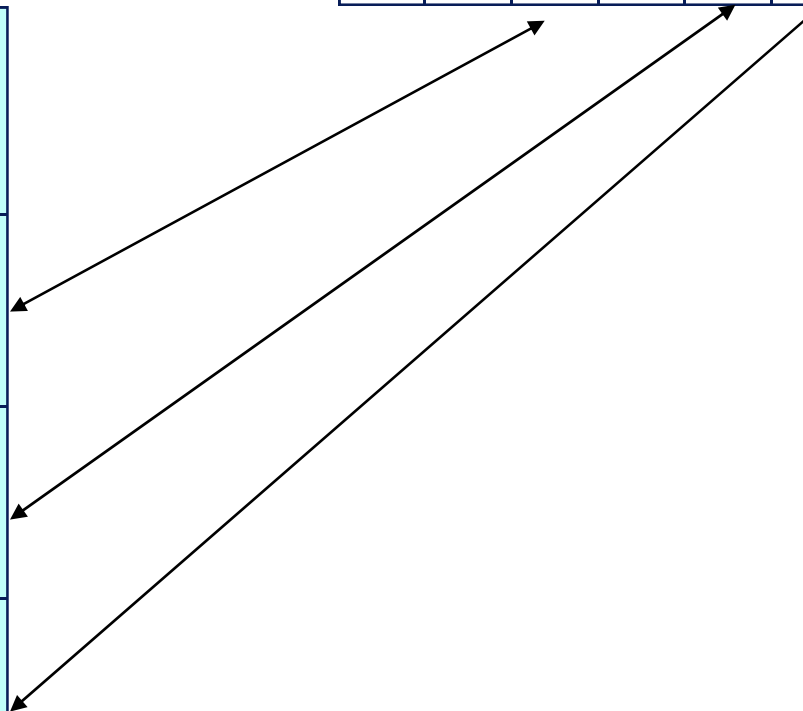
Job Switches e função %SWITCH

*PILHA DE
PROGRAMAS*



JOB SWITCHES

0	1	1	0	0	0	1	0
---	---	---	---	---	---	---	---





Job Switches e função %SWITCH

JOB SWITCHES

0	1	1	0	0	0	1	0
---	---	---	---	---	---	---	---

- 4 Permitem a comunicação Programa-a-Programa dentro de um Job.
- 4 São temporários - Job start/end.
- 4 São inicializados pela **JOB*D.

Default: 00000000

- 4 Devemos utilizar máscaras para testar ou checar as condições:

0 = OFF

1 = ON

X = ignorado

- 4 A função Built-in *%SWITCH(mask)* pode ser usada como uma variável lógica.





Função %SWITCH exemplos

JOB SWITCHES

0	1	1	0	0	0	1	0
---	---	---	---	---	---	---	---

- 1. IF (%SWITCH(10XXXXX0)) CALL IMESPGM1.*
- 2. IF (&BAL > 0) CHGJOB SWS('1XXXXXXXX').*
- 3. CHGVAR &IN90 %SWITCH(00000001).*





Recuperando atributos externos Job Attributes

RTVJOBA E CHGJOB

Pode-se recuperar atributos de JOB colocando-se seus valores em variáveis da linguagem de controle para o controle de aplicações.

Comandos:

RTVJOBA JOB(&WRKSTN)

RTVJOBA USER(&NOME)





Recuperando atributos externos System Values

RTVSYVAL E CHGSYVAL

Pode-se recuperar valores do sistema (*Informações de controle*) para dentro de um programa e manipulá-los como variáveis usando o comando RTVSYVAL.

Exemplos:

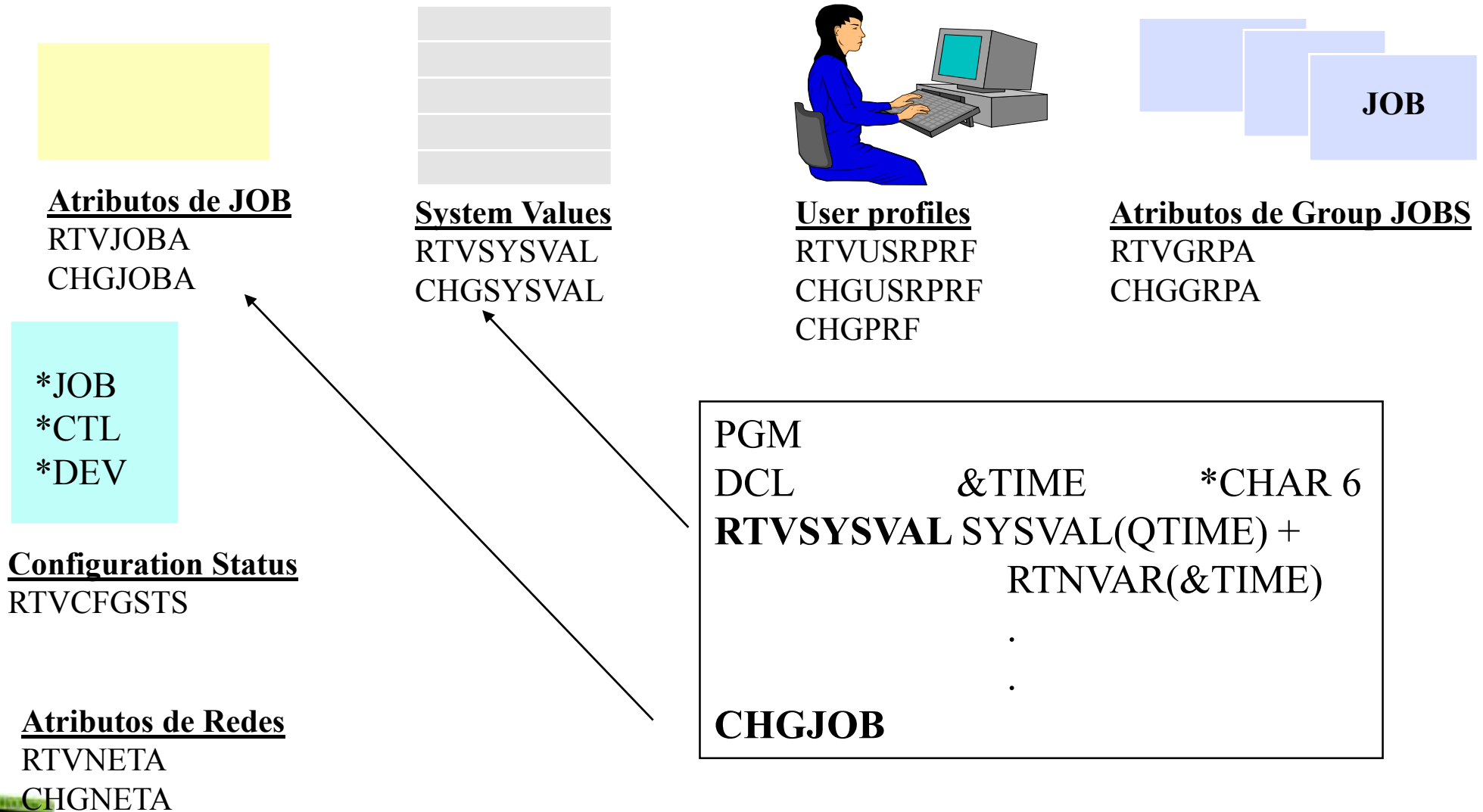
```
RTVSYVAL SYVAL(QTIME) RTNVAR(&TIME)
```

```
RTVSYVAL SYVAL(QDATE) RTNVAR(&DATE)
```





Recuperando atributos externos





EXEMPLO DE RTVJOBA E CHGJOB

PGM

DCL	VAR(&OUTQNAME) TYPE(*CHAR) LEN(10)
DCL	VAR(&OUTQLIB) TYPE(*CHAR) LEN(10)
DCL	VAR(&JOB) TYPE(*CHAR) LEN(10)
RTVJOBA	OUTQ(&OUTQNAME) OUTQLIB(&OUTQLIB) JOB(&JOB)
CHGJOB	OUTQ(IMESLIB/PAYOUTQ)
CALL	PGM(PAYCHECKS)
CHGJOB	OUTQ(&OUTQLIB/&OUTQNAME)
SNDMSG	MSG(' Checks enviados para Output Queue PAYOUTQ.') + TOMSGQ(&JOB)
ENDPGM	





CONVERSÃO DE DATAS CVTDAT

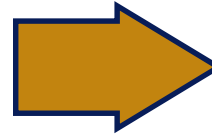
CVTDAT

DATE(Constante ou variável)

TOVAR(Variável CL)

FROMFMT

***JOB
*SYSVAL
*MDY
*DMY
*YMD
*JUL
etc.**



TOFMT

***JOB
*SYSVAL
*MDY
*DMY
*YMD
*JUL
etc.**

TOSEP

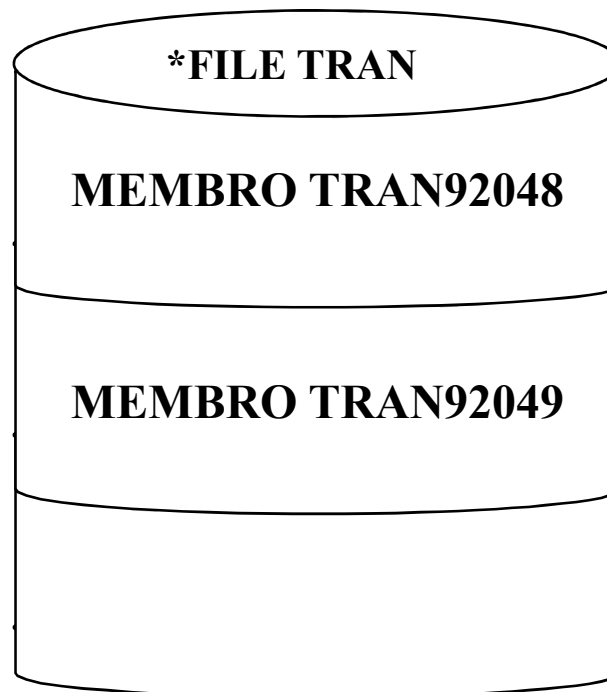
***JOB
*SYSVAL
*NONE
Separator
Character**





Exemplo – Conversão de datas CVTDAT

Geração de arquivo com membros separados para as transações diárias





Exemplo– Conversão de datas CVTDAT

Geração de arquivo com membros separados para as transações diárias

```
PGM
DCL      &MBR          *CHAR  LEN(10)
DCL      &SYSDATE      *CHAR  LEN(6)
DCL      &JUL          *CHAR  LEN(5)

RTVSYSVAL      QDATE RTNVAR(&SYSDATE)

CVTDAT DATE(&SYSDATE) FROMFMT(*SYSVAL) +
        TOVAR(&JUL) TOFMT(*JUL) +
        TOSEP(*NONE)

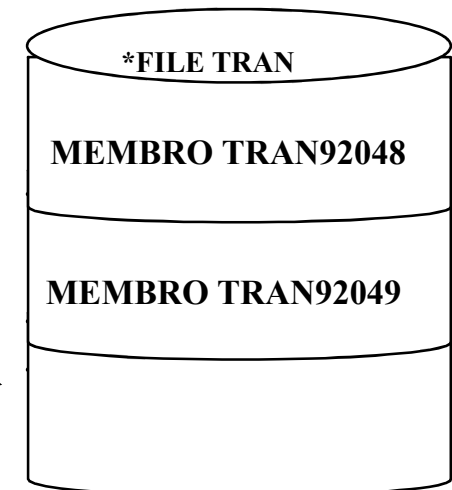
CHGVAR      VAR(&MBR) VALUE('TRAN' *CAT &JUL)

ADDPFM      FILE(TRAN)      MBR(&MBR)

OVRDBFFILE(TRAN)      MBR(&MBR)

CALL      PGM(PAYTRAN)

ENDPGM
```





Usando o Prompt dentro de um programa CL

Send Message(SNDMSG)

Type choices, press Enter.

Message Text..._____

To user profile...._____ Name, *SYSOPR, *ALLACT

Bottom

F3=Exit F4=Prompt F5=Refresh F10=Additional parameters

F12=Cancel F13=How to use this display F24=More Keys





Usando o Prompt dentro de um programa CL

Se um comando é codificado sem nenhum parâmetro, porém precedido de um sinal de interrogação (?), o usuário receberá um PROMPT para os parâmetros do comando durante a execução do programa.

? SNDMSG





PROMPT – Programa especifica alguns valores de parâmetros

?SNDMSG TOMSGQ(QSYSOPR)

Send Message(SNDMSG)

Type choices, press Enter.

Message Text... _____

To user profile.... _____ Name, *SYSOPR, *ALLACT...
Additional Parameters

To message queue... > **QSYSOPR** Name, *SYSOPR
***LIBL** Name, *LIBL, *CURLIB

PROTEGIDO Bottom

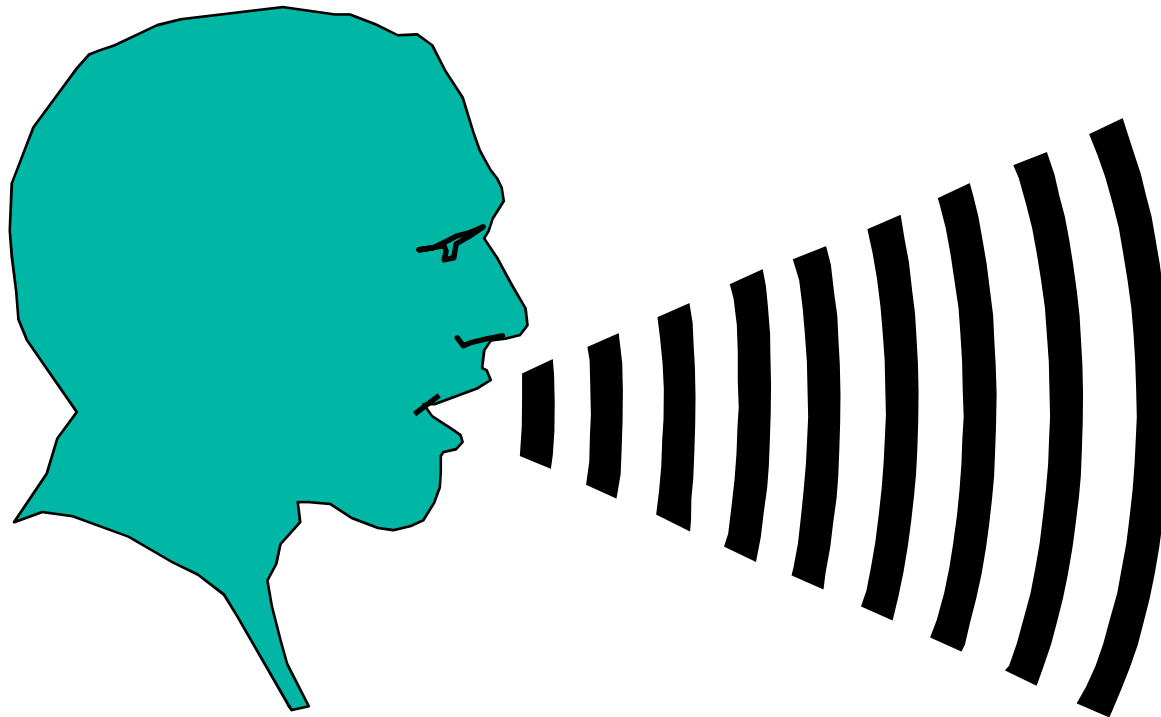
F3=Exit F4=Prompt F5=Refresh F10=Additional parameters
F12=Cancel F13=How to use this display F24=More Keys





Capítulo 5

Mensagens





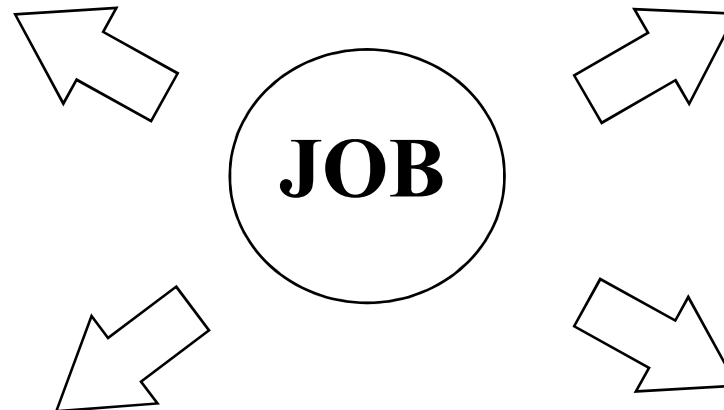
Comunicação básica fila de mensagens

FILA DE MENSAGENS
DA WORKSTATION

FILA DE MENSAGENS
DO USUÁRIO

DSP01
***MSGQ**

USUÁRIO 1
***MSGQ**



USUÁRIO 2
***MSGQ**

QSYSOPR
***MSGQ**

FILA DE MENSAGENS
DA WORKSTATION

FILA DE MENSAGENS DO
OPERADOR DO SISTEMA



Fila de mensagens do Usuário e da Workstation

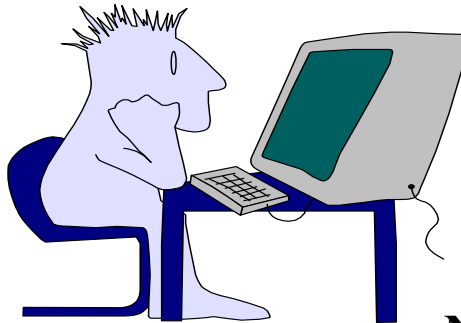
FILA DE MENSAGENS DO USUÁRIO

**USUÁRIO 1
*MSGQ**

NO SIGNON

- Alocadas para o Job do Usuário.
- DELIVERY MODE (Modo de Recepção)
Notify (Notificação)
- Mensagens que estavam armazenadas (em Hold) são enviadas, ligando-se indicadores visual e sonoro de mensagem recebida.

NOTIFICAÇÃO



NO SIGNOFF

FILA DE MENSAGENS DA WORKSTATION

**DSP01
*MSGQ**

- Desalocadas no Job do Usuário.
- DELIVERY MODE (Modo de recepção)
Hold (Armazenamento de mensagens na fila, sem envio para o terminal).
- Novas mensagens recebidas ficam armazenadas.



Mensagens são sempre enviadas para Filas de Mensagens

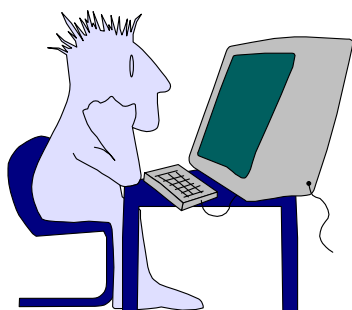
FILA DE MENSAGENS
DO USUÁRIO

USUÁRIO 1
*MSGQ

FILA DE MENSAGENS
DA WORKSTATION

DSP01
*MSGQ

SENDMSG



JOSÉ
*MSGQ

QSYSOPR
*MSGQ

FILA DE MENSAGENS
DO USUÁRIO

FILA DE MENSAGENS DO
OPERADOR DO SISTEMA

*EXT

External Message
Queue

QCMD

Fila de mensagens
do programa

PGMA

Fila de mensagens
do programa

PGMB

Fila de mensagens
do programa





Mensagens podem ser

PRÉ-DEFINIDAS

ou

IMEDIATAS

Linguagem de Controle:

SNDUSRMSG
SNDPGMMSG
RTVMSG

Display Files:

ERRMSGID
SFLMSGID
MSGCON
MSGID

Linguagem de Controle:

SNDUSRMSG
SNDPGMMSG
RTVMSG

Display Files:

ERRMSG
SFLMSG





SNDMSG

LINHA DE COMANDOS



SNDMSG MSG ('USCS - CIÊNCIA DA COMPUTAÇÃO')
TOMSGQ (ABC)

PROGRAMA CL



```
PGM
DCL &MSG *CHAR 10 ABC
SNDMSG MSG ('IMES - CIÊNCIA DA COMPUTAÇÃO') +
        TOMSGQ(&MSG)
ENDPGM
```





Enviando mensagens tipo break (*SNDBRKMSG*)

Você pode enviar mensagens do tipo “BREAK” para um JOB usando o comando SNDBRKMSG. Este comando deve ser usado sempre que você estiver enviando uma mensagem que requer atenção imediata do operador de uma estação de trabalho.

PALAVRAS CHAVES (KEYWORDS)

MSG	Texto da mensagem ou nome da variável que contenha a mensagem.
TOMSGQ	Fila de mensagens para onde a mensagem será enviada.
MSGTYPE	Tipo de mensagem *INFO ou *INQ.
RPYMSGQ	Fila de mensagem que irá receber a resposta caso requerida.





Interação com o usuário

**Espera pela
resposta**

A fita está montada? (S/N)

Type reply, press Enter.

Reply... y

F3=Exit F12=Cancel

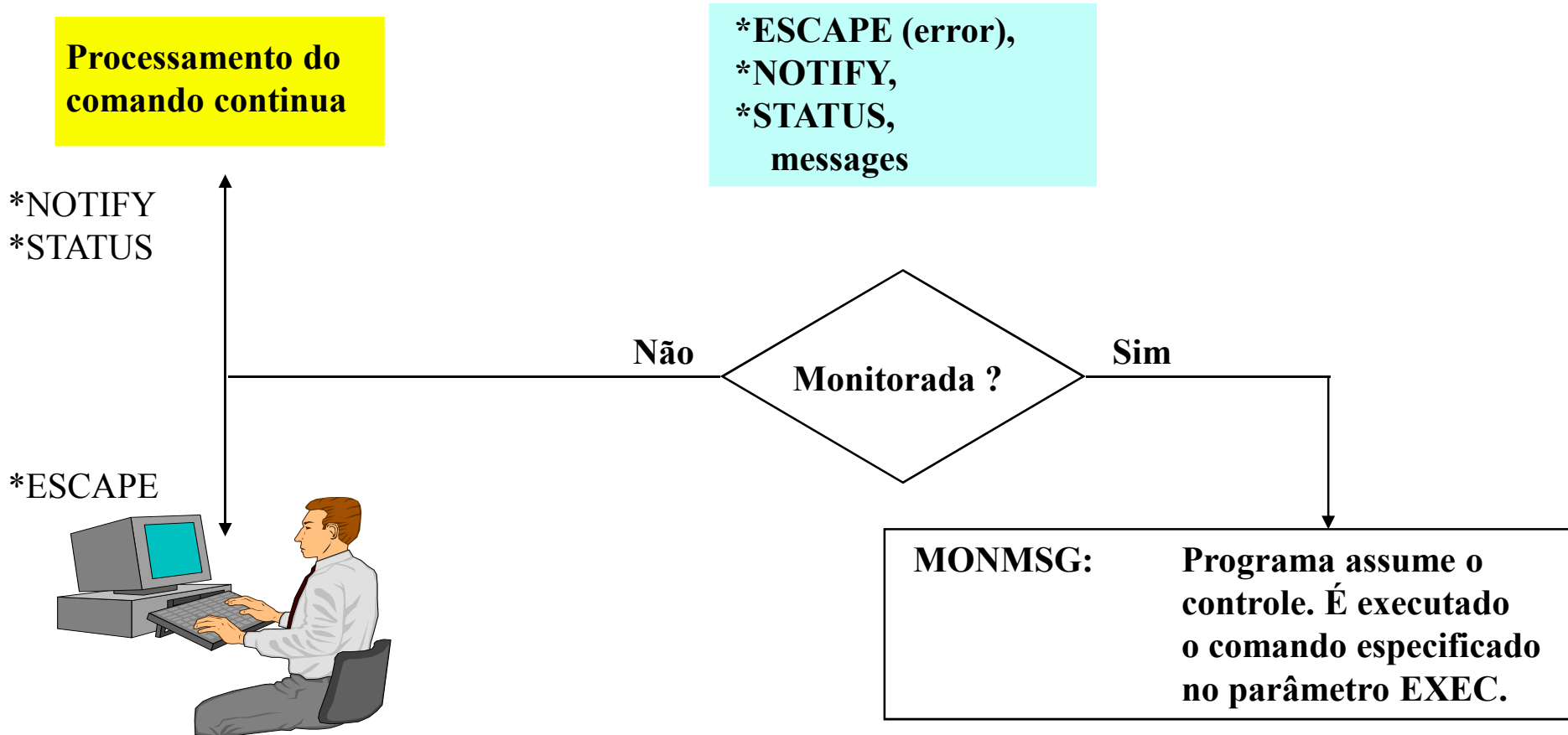
*EXT
Message
Queue

```
PGM                /* Save da biblioteca IMESLIB */
DCL    VAR(&RESPOSTA) TYPE(*CHAR) LEN(1)
SENDUSRMSG  MSG('A fita está montada? (S/N)') +
                VALUES (S N) DFT(N) +
                MSGRPY(&RESPOSTA) +
                TRNTBL(QSYSTRNTBL) +
                MSGTYPE(*INQ) +
                TOMSGQ(*)
IF      COND(&RESPOSTA = S) THEN +
                (SAVLIB LIB(IMESLIB) DEV(TAP01))
ENDPGM
```





Manuseio de exceções: Usuário x Programa



R - Repete o comando
I - Continua para o próximo comando
C - Cancela o programa. Function check CPF9999
enviado para o programa chamador.





Comando MONMSG disponibilidade de arquivo
(Exemplo)

```
PGM
ALCOBJ          OBJ(IMESLIB/PAYMAST *FILE *EXCL) WAIT(0)
MONMSG         MSGID(CPF1002) EXEC(DO)
                 SNDUSRMSG MSG('Programa não pode rodar, pois +
                 o arquivo PAYMAST não está disponível.') +
                 MSGTYPE (*INFO) TOMSGQ (*EXT)

                 RETURN
                 ENDDO
CALL            PGM(IMESPGM)
DCLOBJ          OBJ ((IMESLIB/PAYMAST) *FILE *EXCL)
ENDPGM
```





Exemplo - Comando MONMSG F3 durante o comando Prompting

PGM

DCL &COUNT *DEC

**.
. .
.**

SNDMSG: ?SNDMSG

MONMSG MSGID(CPF6801) EXEC(DO) /*F3=EXIT */
CHGVAR VAR(&COUNT) VALUE(&COUNT + 1)
IF COND(&COUNT = 3) THEN (RETURN)
ELSE CMD(GOTO SNDMSG)

ENDDO

**.
. .
.**





Comando MONMSG Monitoramento de Mensagens

MONMSG MSGID() CMPDTA() EXEC()

Parâmetro MSGID - Requerido

MSGID(MCH1211) Somente esta mensagem

MSGID(CPF1600) CPF1601 - CPF1699

MSGID(CPF0000) CPF0001 - CPF9999

MSGID(CPF9999) Function Check:
Serão ativadas pelas mensagens de
Escape não monitoradas





Comando MONMSG
Monitoramento de mensagens

MONMSG MSGID() CMPDTA() EXEC()

Parâmetro *EXEC* - Opcional

Exemplo:

**MONMSG MSGID(CPF2182) +
*EXEC(GOTO Roterro)***





Monitorando mensagens

Formato das mensagens (CÓDIGO)

PPPMNN

PPPM00

PPP0000

Exemplos:

MONMSG MSGID(CPF2200) EXEC(GOTO ROTFIM)

MONMSG MSGID(RPG0000) EXEC(GOTO FIMRPG)

MONMSG MSGID(CPF2105 MCH1211) EXEC(GOTO ROTFINAL)

MONMSG MSGID(CPF2105)





MONMSG: Nível de programa, Nível de Comando





Monitoramento de mensagens Check de existência de objetos

CHKOBJ **OBJ(nome-do-objeto)** **+**
 OBJTYPE(tipo-do-objeto) **+**
 MBR(*NONE | *FIRST | membro-de-arquivo)

Exemplo:

CHKOBJ **OBJ(IMESLIB/APR01)** **+**
 OBJTYPE(*PGM)

/* **CHECAGEM DA EXISTÊNCIA DO OBJETO** ***/**

MONMSG.....





Monitoramento de mensagens Check de existência de objetos e autorização

CHKOBJ **OBJ(nome-do-objeto)** **+**
 OBJTYPE(tipo-do-objeto) **+**
 MBR(*NONE | *FIRST | membro-de-arquivo)+
 AUT(direitos-de-autorização-sobre-o-objeto)

Exemplo:

CHKOBJ **OBJ(IMESLIB/APPVEND)** **+**
 OBJTYPE(*FILE) **+**
 MBR(APPVEND) AUT(*DLT)

/* CHECAGEM DA EXISTÊNCIA DO OBJETO */
/* E AUTORIDADE */

MONMSG.....





Monitoramento de mensagens *Exemplo*

```
PGM
/* MONMSG a nível de Programa */
MONMSG CPF9801 EXEC(GOTO error)
/* Checagem de arquivos */
CHKOBJ PAYMAST *FILE
CHKOBJ GLMAST  *FILE
/* Criação de um novo arquivo de trabalho */
AGAIN: CRTPF QTEMP/WORK
MONMSG CPF7302 EXEC(DO)
        DLTF QTEMP/WORK
        GOTO AGAIN
        ENDDO

/* Criação de um novo arquivo PAYTRAN */
/* Caso o mesmo não exista */
CRTPF PAYLIB/PAYTRAN
CALL PAYMONTHLY
RETURN
ERROR: SNDUSRMSG MSG('Arquivo não existe. +
(a) Check *LIBL para PAYLIB          +
(b) Chame Supervisor.'               +
MSGTYPE(*INFO)
ENDPGM
```

PAYLIB:

PAYMAST
(Deve existir)

GLMAST
(Deve existir)

PAYTRAN
(Arquivo a ser criado
ou usa existente)

QTEMP:

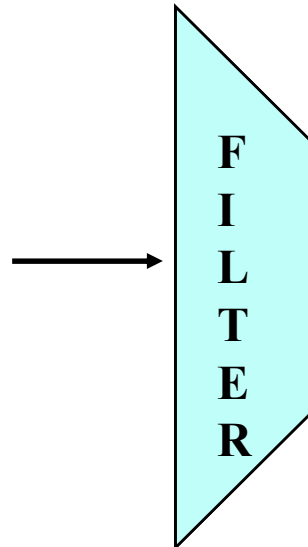
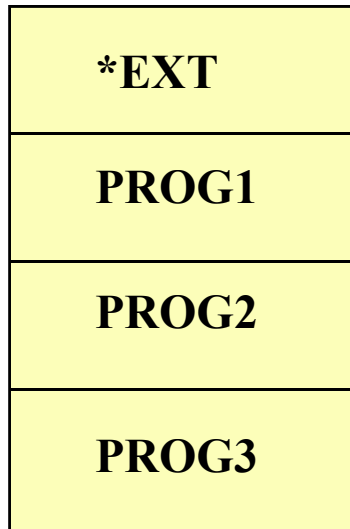
PAYTRAN
(Arquivo novo a cada
execução)



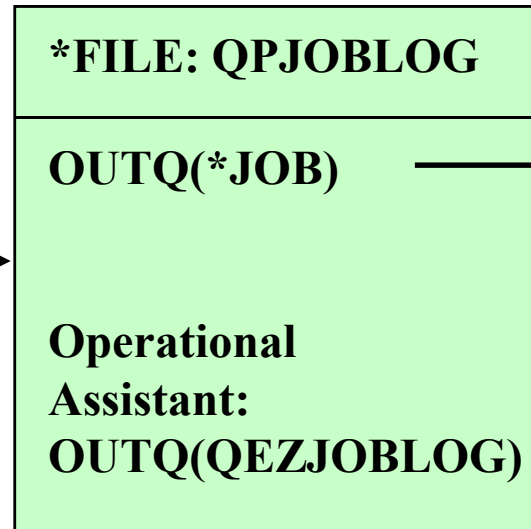


JOB LOG

JOB: DSP01



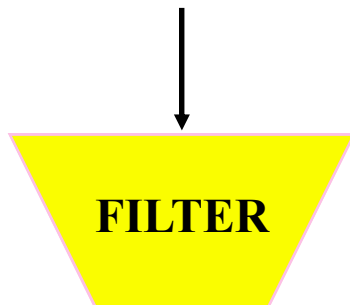
EOJ



**CRTOUTQ OUTQ(JOBLOG)
CHGPRTF FILE(QPJOBLOG)
OUTQ(JOB)**

***OUTQ: JOBLOG**

***OUTQ:
QUSRSYS/QEZJOBLOG**





JOB LOG

FILTER

CRTCLPGM...LOG(*JOB/*YES/*NO)

Comandos dentro
de Programas CL

**CHGJOB
CHGJOB
CRTJOB**

LOG

message level	message severity	message text level
------------------	---------------------	--------------------------

...LOGCLPGM(*YES/*NO)

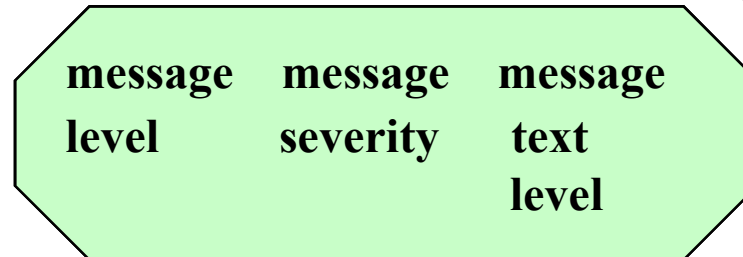




CHGJOB
CHGJOB
CRTJOB

LOG

JOB LOG



FILTER

...LOGCLPGM(*YES/*NO)

Message-level

Message-severity

0 - Nenhuma informação é logada		0-99
1 - Mensagens *EXT se Severity *GE	—————→	0-99
2 - Level 1 Comandos e suas mensagens se Severity *GE	—————→	0-99
3 - Level 1 Comandos Mensagens dos comandos se Severity *GE	—————→	0-99
4 - Comandos Todas as mensagens se Severity *GE	—————→	0-99



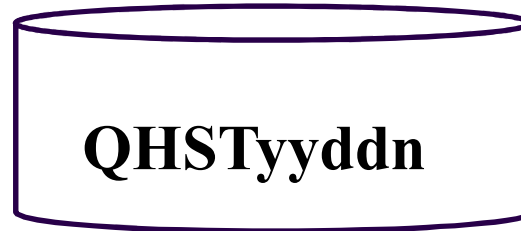
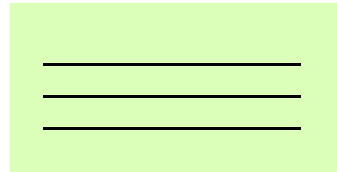


QHST HISTORY LOG

QHST *MSGQ

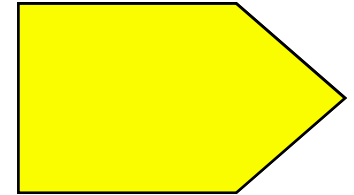
System
Subsystem
Job Start/Completion
Device Status
QSYSOPR messages

DSPLOG



QHSTyddn

“Log Version”



QHSTLOGSIZ

As mensagens são gravadas da Message Queue QHST para a Current Log Version quando:

- A Message Queue QHST estiver cheia

- DSPLOG estiver executando

Obs.: QHSTLOGSIZ é uma “SYSTEM VALUE”





ARQUIVO QHSTyyddn

- **Pode ser processado por programas de usuário**
 - . *Performance reporting*
 - . *Job Accouting*

- **O usuário se responsabiliza pela deleção de “Old Files”**
 - . *WRKF FILE(QSYS/QHST*)*
 - . *Use Operational Assistant*

Observações:

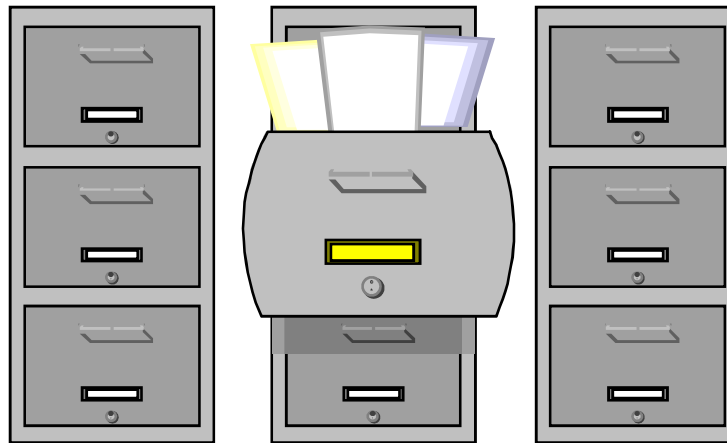
1. O valor “yyddd” do nome da versão Log é a data Juliana.
2. O “n” do nome é uma sequência de caracteres(0-9, A-z)





Capítulo 6

Acesso a Banco de Dados





Usando um Data Base File

PGM

DCLF ...

DSPxxx ... OUTPUT(*OUTFILE) +
 OUTFILE (MYFILE) +
 OUTMBR(*FIRST *REPLACE)

READ: **RCVF** ←
 MONMSG CPF0864 EXEC(RETURN)

GOTO READ





Declarando o Data Base file

***PGM**

DCLF FILE(IBM-file)

DSPxxx OUTPUT(*OUTFILE) +
 OUTFILE(user-file) +
 OUTMBR(*FIRST *REPLACE)

OVRDBF FILE(IBM-file) TOFILE(user-file)

RCVF





Declarando o Data Base file

**DSPxxx... OUTPUT(*OUTFILE) +
 OUTFILE(user-file) +
 OUTMBR(*FIRST *REPLACE)**

***PGM**

DCLF FILE(user-file)

**DSPxxx OUTPUT(*OUTFILE) +
 OUTFILE(user-file) +
 OUTMBR(*FIRST *REPLACE)**

RCVF





Operações com o Data Base file

4Declare File, DCLF

- PF ou LF
- Um por programa

4Open File

- Primeiro RCVF
- Somente Input

4Read Next Record: RCVF

- EOF message: CPF0864

4File Closed

- EOF
- RETURN, TFRCTL





Uso de Output File a partir de um comando Display

PGM	PARM(&USER LIB)	
DCL	&USER	*CHAR 10
DCL	&LIB	*CHAR 10
1	DCLF QSYS/QADSPOBJ	

- O arquivo declarado, *QADSPOBJ* na QSYS, é um arquivo fornecido pela IBM.
- Este arquivo é referenciado pelo compilador CL para determinar o formato dos registros e para declarar variáveis para os campos do formato do registro.





Uso de Output File a partir de um comando Display

PGM	PARM(&USER LIB)	
DCL	&USER	*CHAR 10
DCL	&LIB	*CHAR 10
DCLF	QSYS/QADSPOBJ	

2

DSPOBJD	OBJ(&LIB/*ALL)	+
	OBJTYPE(*FILE *PGM *DTAARA)	+
	OUTPUT(*OUTFILE)	+
	OUTFILE(QTEMP/DSPOBJD)	

- O comando DSPOBJD cria um arquivo chamado DSPOBJD na biblioteca QTEMP.
- Este arquivo tem o mesmo formato do arquivo QADSPOBJ.





Uso de Output File a partir de um comando Display

```
PGM          PARM(&USER LIB)
DCL          &USER          *CHAR 10
DCL          &LIB            *CHAR 10
DCLF         QSYS/QADSPOBJ
DSPOBJD      OBJ(&LIB/*ALL)      +
              OBJTYPE(*FILE *PGM *DTAARA) +
              OUTPUT(*OUTFILE)      +
              OUTFILE(QTEMP/DSPOBJD)
```

3

```
OVRDBF      QADSPOBJ      +
            TOFILE(QTEMP/DSPOBJD)
```

- O comando OVRDBF (Override Data Base) substitui o arquivo declarado no comando DCLF (QSYS/QADSPOBJ) pelo arquivo DSPOBJD na biblioteca QTEMP.





Uso de Output File a partir de um comando Display

PGM	PARM(&USER LIB)	
DCL	&USER	*CHAR 10
DCL	&LIB	*CHAR 10
DCLF	QSYS/QADSPOBJ	
DSPOBJD	OBJ(&LIB/*ALL)	+
	OBJTYPE(*FILE *PGM *DTAARA)	+
	OUTPUT(*OUTFILE)	+
	OUTFILE(QTEMP/DSPOBJD)	
OVRDBF	QADSPOBJ	+
	TOFILE(QTEMP/DSPOBJD)	

4

READ: RCVF

- O comando RCVF efetua a leitura de um registro no arquivo DSPOBJD.
- Os valores dos campos do registro são copiados nas correspondentes variáveis CL, as quais são implicitamente declaradas pelo comando DCLF.





Uso de Output File a partir de um comando Display

```
PGM          PARM(&USER LIB)
DCL          &USER          *CHAR 10
DCL          &LIB            *CHAR 10
DCLF         QSYS/QADSPOBJ
DSPOBJD      OBJ(&LIB/*ALL)      +
                   OBJTYPE(*FILE *PGM *DTAARA)  +
                   OUTPUT(*OUTFILE)             +
                   OUTFILE(QTEMP/DSPOBJD)
OVRDBF       QADSPOBJ          +
                   TOFILE(QTEMP/DSPOBJD)
READ:        RCVF
```

5

```
MONMSG      CPF0864      EXEC(RETURN)
/* EXIT QUANDO OCORRER FIM DE ARQUIVO */
```

- A mensagem CPF0864 é monitorada.
- Isto indica que ao ocorrer fim de arquivo o controle é devolvido para o programa chamador.





Uso de Output File a partir de um comando Display

```
PGM          PARM(&USER LIB)
DCL          &USER          *CHAR 10
DCL          &LIB           *CHAR 10
DCLF         QSYS/QADSPOBJ
DSPOBJD      OBJ(&LIB/*ALL)          +
          OBJTYPE(*FILE *PGM *DTAARA)      +
          OUTPUT(*OUTFILE)                +
          OUTFILE(QTEMP/DSPOBJD)
          OVRDBF      QADSPOBJ          +
          TOFILE(QTEMP/DSPOBJD)
```

READ: RCVF

MONMSG CPF0864 EXEC(RETURN)

/* EXIT QUANDO OCORRER FIM DE ARQUIVO */

6

```
GRTOBJAUTOBJ(&ODLBNM/&ODOBTP)+
          USER(&USER) AUT(*CHANGE)
```

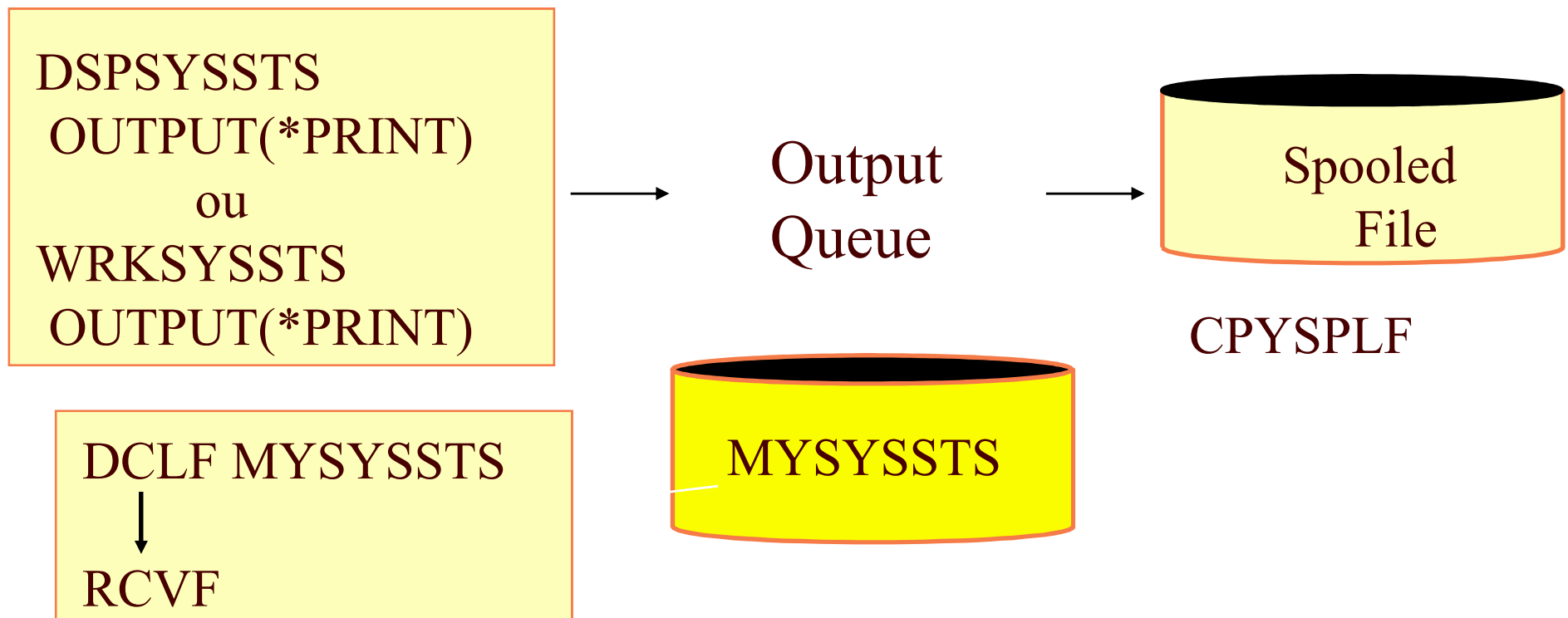
- O comando GRTOBJAUT é processado, usando as variáveis para o nome do objeto, nome da biblioteca e tipo, os quais foram lidos pelo comando RCVF.





COMANDOS SEM O PARÂMETRO OUTFILE

Exemplo: Processar a saída dos comandos WRKSYSSTS ou DSPSYSSTS





Capítulo 7

Jobs Batch





Visão do Usuário - Job Batch

JOB(JOB1 JOB1)

CMD(CALL PGM1)

JOB(JOB1)

SBMJOB



Memória
Principal



Output
Queue

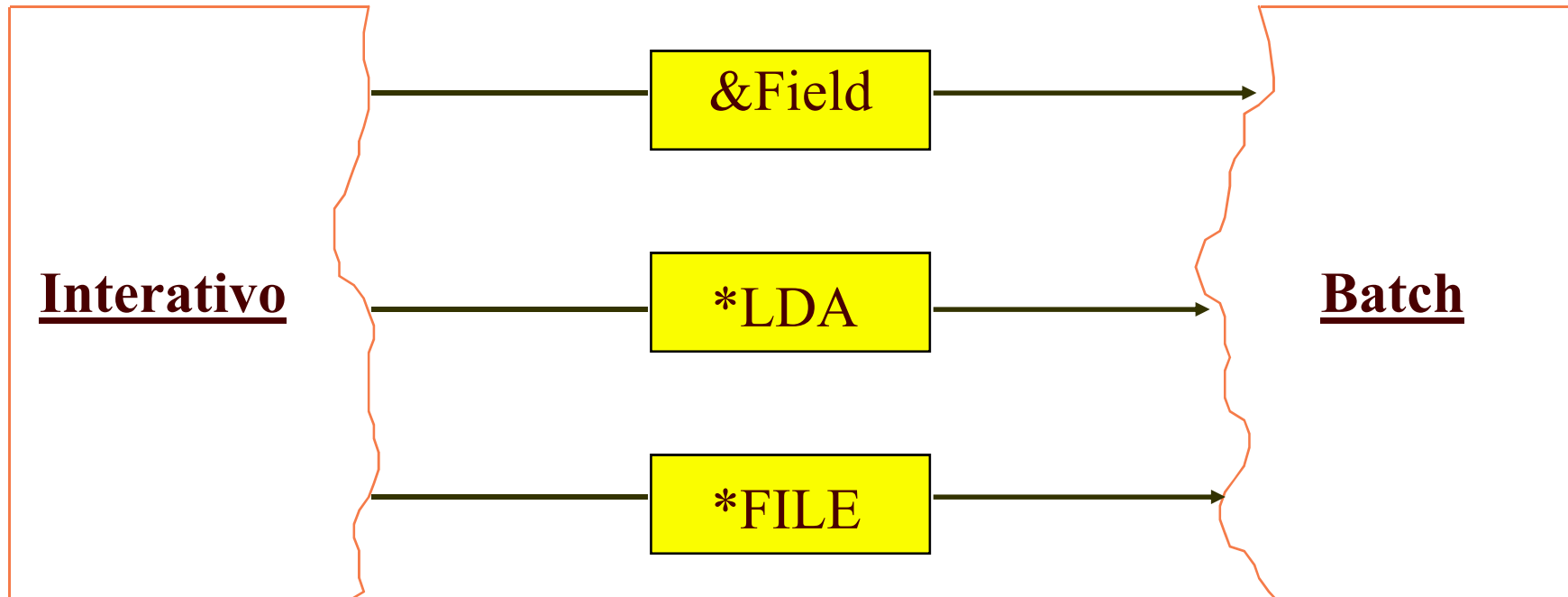
JOB1

ARQUIVO
DE
DADOS

SPLF

RELATÓRIO

Job Batch



Se uma função Batch é removida de programas interativos, haverá melhor desempenho na execução do programa interativo.



COMANDO SBMJOB

*LDA
Order #
12345

*LDA
Order #
12345

*LDA
Order #
12345

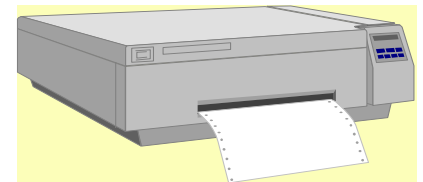
PROGRAMA
INTERATIVO

SUBMIT JOB

PROGRAMA
BATCH

Fim da
ordem

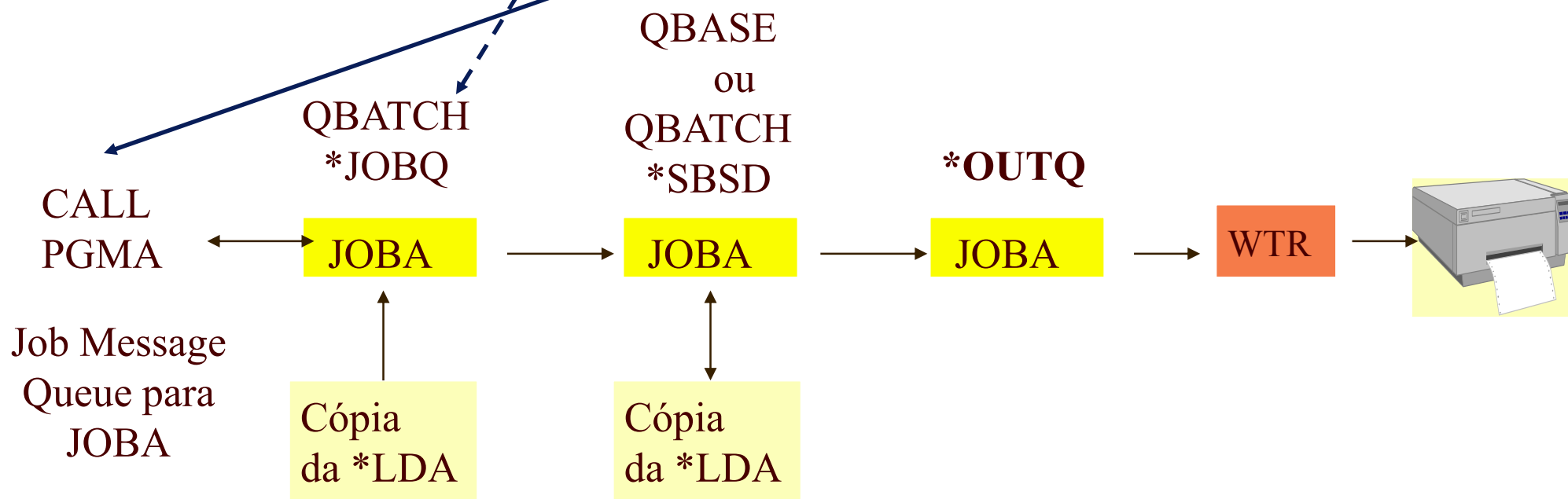
ORDER
FILE





COMANDO SBMJOB

SBMJOB JOB(JOBA) CMD(CALL PGMA)





PARÂMETROS DO SBMJOB

SBMJOB CMD(command) + Parâmetros abaixo

Default para *CURRENT

PRTDEV
OUTQ
USER
PRTTXT
SYSLIBL
CURLIB
INLLIBL
SCDDATE
SCDTIME

Default para *JOB

JOB
JOBQ
JOBPTY
OUTPTY
LOG
LOGCLPGM
INQMSGRPY
DATE
SWS
HOLD

Outros Defaults do SBMJOB

JOB(*USRPRF)
RTGDTA(QCMDB)
RQSDTA(*CMD)
DSPSBMJOB(*YES)
MSGQ(*USRPRF)





EXEMPLO DE SBMJOB

**LDA*

JOB INTERATIVO

54321SHIP VIA AIR

ORDERS

&INVNA

&SHIP

PGM

DCL &SHIP *CHAR 15

DCL &INVNA *CHAR 5

DCL &INVN *DEC (5 0)

DCL &TIME *CHAR 6

⋮

CHGVAR &INVNA &INVN

CHGDTAARA (*LDA(1 5)) &INVNA

CHGDTAARA (*LDA(6 15)) &SHIP

RTVSYSVAL QTIME &TIME

SBMJOB CMD(CALL BATCHINV PARM(&TIME))

⋮





EXEMPLO DE SBMJOB

Cópia da *LDA

JOB BATCH

54321SHIP VIA AIR

&INVNA

&SHIP

BATCHINV

```
PGM  PARM(&TIME)
DCL  &SHIP      *CHAR  15
DCL  &INVNA     *CHAR    5
DCL  &INVN      *DEC    (5 0)
DCL  &TIME      *CHAR    6
      :
      :
RTVDTAARA (*LDA(1 5)) RTNVAR(&INVNA)
RTVDTAARA (*LDA(6 15)) RTNVAR(&SHIP)
CHGVAR      &INVN&INVNA
      :
```

