

## Раздел 8

### Секция File Management (управление файлами)



# Оператор NASTRAN

- Оператор NASTRAN входит в секцию FMS и является необязательным.
- Используется для изменения глобальных параметров выполнения расчета. Если этот оператор используется, то располагать его необходимо в первой строке входного файла.
- Используется только в особых случаях и не нужен для большинства расчетов
- Например, для изменения параметра BUFFSIZE (размер буфера), впишите в первую строку либо

**NASTRAN BUFFSIZE=12929**

или

**NASTRAN SYSTEM(1)=12929**

# Оператор NASTRAN (продолжение)

- **Полный список и описание параметров оператора NASTRAN смотрите в *Разделе 1 MSC Nastran Quick Reference Guide*.**

# Рекомендуемые значения BUFFSIZE

Степени свободы	Buffsize (Машинных слов)
DOF < 100,000	8193 (по умолчанию)
100,000 < DOF < 200,000	16385
400,000 < DOF	32769
	макс. значение 65537

- Эти значения были выбраны для представления наилучшего компромисса между скоростью доступа к базе данных и скоростью сохранения информации для обычных задач.
- Размер блоков ввода/вывода (I/O transfer size) определяется из соотношения:

$$(\text{BUFFSIZE} - 1) \times \text{BPW},$$

где BPW равно 8 для UNICOS и 4 для других систем.

(См. *Installation and Operations Guide; V2005, страница 110.*)

- **Пример:**

(*MSC Nastran Quick Reference Guide: Volume 1; V2005, страница 13*)

**NASTRAN SYSTEM (1) = 4097**

ИЛИ

**NASTRAN BUFFSIZE = 4097**

# Секция FILE MANAGEMENT (FMS)

- Секция FMS является необязательной и нет необходимости ее использования для большинства задач.
- Используется для инициализации или присоединения существующих баз данных MSC Nastran и FORTRAN файлов.
- Команды FMS одинаково работают на любых платформах.

# Секция FILE MANAGEMENT (продолжение)

- **Типичные применения**
  - Инициализация и расширение баз данных (имя, размер, размещение и т.д.).
  - Разделение большой базы данных на несколько меньших, которые позволяют проводить решение больших задач на машинах с ограниченным дисковым пространством.
  - Расширение существующей базы данных.
  - Управление рестартами.
- **Для более детального описания FMS смотрите**  
*Раздел 2 MSC.Nastran Quick Reference Guide*

# Обзор FMS

- **Обеспечивает динамическое распределение файлов пользователя**
  - Создание/подсоединение файлов базы данных
  - Создание/подсоединение FORTRAN файлов
- **Обеспечивает манипуляции с данными**
  - Рестарт
  - Инициализация и расширение базы данных
  - Управление данными в базе данных
  - Печать каталогов базы данных
- **Обеспечивает связь с внешними базами данных**

# База данных

- **DBSET** – Набор файлов которые используются программой в тех или иных целях.
- База данных MSC Nastran состоит из нескольких DBSETов, которые автоматически создаются программой:
- **MASTER** – Управляющая директория – содержит описание базы данных, список всех файлов, содержащихся во всех DBSETах, используемых при запуске и таблицы содержания ("table of contents") для каждого DBSETа.
- **DBALL** – Постоянные данные.

# База данных (продолжение)

- **SCRATCH** – Временные рабочие файлы для временных блоков данных, автоматически удаляются после завершения расчета.
- **SCR300** – Временные рабочие файлы для модулей, автоматически удаляются после завершения расчета.

# Необходимые DBSETы

- **MASTER и DBALL должны сохраняться, если база данных будет использоваться для последующих рестартов. Если рестарт не планируется, то они могут быть удалены после расчета.**
- **Для автоматического удаления MASTER и DBALL используйте PARAM, DBALL, SCRATCH или команду `scr = yes`**

# Типичные генерируемые файлы

- **Файлы генерируемые при обработке входного файла test.dat:**

test.MASTER

test.DBALL

test.f04

test.f06

test.log

test.pch

test.plt

test.op2

test.xdb

- **Файлы test.pch (вспомогательный файл),  
test.plt (файл графопостроителя),  
test.op2 (информация для MSC Patran),  
test.xdb (xdb файл)**

**создаются по запросам пользователя.**

# Типичные генерируемые файлы (продолжение)

- Если не планируются рестарты или работа с базой данных, то пользователь может указать, что файлы MASTER и DBALL следует помещать во временный каталог. Тогда они будут автоматически уничтожены после окончания расчета.
- Пример:

```
nastran test scr=yes
```

# Рестарты

- **Структурные решения (SOL's 101,103 и т.д.)**
  - Включают возможность автоматического рестарта
  - Минимум входных данных для рестарта
  - Программа определяет изменения в модели и реагирует на них

# Автоматические рестарты (SOL 101-200)

- Каждый рестарт создает новую версию базы данных. Предыдущая версия, используемая для рестарта, удаляется после окончания расчета, если пользователь не указал KEEP в записи RESTART.
- Вся секция BULK DATA сохраняется в базе данных. Во входном файле для рестарта требуются только изменения в секции BULK DATA. Новые записи секции BULK DATA сливаются с записями секции BULK DATA в базе данных.
- Все команды секции CASE CONTROL, управляющие решением должны быть в файле рестарта.
- Для рестарта программа сравнивает содержание секций CASE CONTROL и BULK DATA с их состоянием до рестарта, а затем решает, что необходимо рассчитать или пересчитать в данной задаче.

# Автоматические рестарты (продолжение)

- Для решения задачи выполняются только необходимые операции.
- Рестарты вызываются с помощью оператора RESTART находящегося в секции FMS.

# Пример рестарта

- Рассмотрим рестарт на примере консольной пластины (Пример 5)
- База данных должна быть сохранена при первом запуске на расчет (холодный старт).
  - То есть, команда должна выглядеть так (предполагаемое имя файла SOLN5.DAT):

```
nastran soln5 scr=no
```

# Часть файла F04 (холодный старт)

```

1
MACHINE                MODEL                OPERATING SYSTEM      MSC.Nastran          BUILD DATE           RUN D
Intel                  PentiumIII/550 (TINM Windows 2000 5.0 (Bu  VERSION 2001.0      OCT 12, 2000       OCT 1'

```

```

=== MSC.Nastran EXECUTION SUMMARY ===

```

Day_Time	Elapsed	I/O_Mb	Del_Mb	CPU_Sec	Del_CPU	Subroutine
17:16:35	0:00	0.0	0.0	0.0	0.0	SEMTRN BGN
17:16:35	0:00	0.0	0.0	0.0	0.0	SEMTRN END
17:16:35	0:00	0.0	0.0	0.0	0.0	DBINIT BGN

```

** CURRENT PROJECT ID = 'BLANK' ** CURRENT VERSION ID = 1

```

```

SUMMARY OF FILE ASSIGNMENT FOR THE PRIMARY DATABASE (DBSNO :

```

ASSIGNED PHYSICAL FILE NAME (/ORIGINAL)	LOGICAL NAME	DBSET	STATUS	BUFFSIZE	CLUSTER SIZE	T
./soln5.MASTER	MASTER	MASTER	NEW	8193	1	00:
./soln5.DBALL	DBALL	DBALL	NEW	8193	1	00:
g:/scratch/soln5.T1588_35.OBJSCR	OBJSCR	OBJSCR	NEW	8193	1	00:
**** MEM FILE ****						
g:/scratch/soln5.T1588_35.SCRATCH	SCRATCH	SCRATCH	NEW	8193	1	00:
g:/scratch/soln5.T1588_35.SCR300	SCR300	SCRATCH	NEW	8193	1	00:
17:16:35	0:00	0.0	0.0	0.0	0.0	DBINIT END
17:16:35	0:00	0.0	0.0	0.0	0.0	XCSA BGN

```

SUMMARY OF FILE ASSIGNMENT FOR THE DELIVERY DATABASE (DBSNO :

```

ASSIGNED PHYSICAL FILE NAME (/ORIGINAL)	LOGICAL NAME	DBSET	STATUS	BUFFSIZE	CLUSTER SIZE	T
k:/nast2001/msc2001/i386/SSS.MASTERA	MASTERA	MASTER	OLD	8193	1	00:
./SSS.MASTERA						
k:/nast2001/msc2001/i386/SSS.MSCOBJ	MSCOBJ	MSCOBJ	OLD	8193	1	00:
./SSS.MSCOBJ						
k:/nast2001/msc2001/i386/SSS.MSCSOU	MSCSOU	MSCSOU	OLD	8193	1	00:
./SSS.MSCSOU						
17:16:36	0:01	34.0	34.0	0.3	0.3	XCSA END
17:16:36	0:01	34.0	0.0	0.3	0.0	CGPI BGN
17:16:36	0:01	34.0	0.0	0.3	0.0	CGPI END
17:16:36	0:01	34.0	0.0	0.3	0.0	LINKER BGN

# Часть файла F04 (холодный старт) (Продолжение)

```
17:16:37 0:02 79.0 0.0 1.2 0.0 SESTATIC311 EXIT BEGN
17:16:37 0:02 79.0 0.0 1.2 0.0 XSEMDR END
```

\*\*\* TOTAL MEMORY AND DISK USAGE STATISTICS \*\*\*

```
+----- SPARSE SOLUTION MODULES -----+
HIWATER      SUB_DMAP      DMAP
{WORDS}     DAY_TIME     NAME      MODULE
1461795    17:16:36    SEKRRS    42  DCMP
```

```
+----- MAXIMUM DISK USAGE -----+
HIWATER      SUB_DMAP      DMAP
{MB}        DAY_TIME     NAME      MODULE
7.563      17:16:37    SESTATIC  311  EXIT
```

\*\*\* DATABASE USAGE STATISTICS \*\*\*

```
+----- LOGICAL DBSETS -----+
DBSET      ALLOCATED  BLOCKSIZE  USED  USED
           {BLOCKS}  {WORDS}   {BLOCKS} %
MASTER      5000      8192      53   1.06
DBALL      250000   8192      90   0.04
OBJSCR      5000      8192      97   1.94
SCRATCH     500100    8192      9   0.00
```

```
+----- DBSET FILES -----+
FILE      ALLOCATED  HIWATER  HIWATER  I/O TRANSFERRE
           {BLOCKS}  {BLOCKS}  {MB}     {GB}
MASTER      5000      53       1.656    0.04
DBALL      250000   90       2.813    0.01
OBJSCR      5000      97       3.031    0.01
{MEMFILE    100      35       1.094    0.00
SCRATCH     250000   1        0.031    0.00
SCR300     250000   1        0.031    0.00
=====
TOTAL:                                0.07
```

# Часть файла F06 (холодный старт)

SUBCASE 1

## D I S P L A C E M E N T V E C T O R

POINT ID.	TYPE	T1	T2	T3	R1	R2	R3
1	G	0.0	0.0	0.0	-2.097525E-02	1.159727E-04	-6.214863E-05
2	G	-3.475199E-05	6.132997E-06	-6.156720E-04	-2.075803E-02	1.145853E-04	-3.997962E-05
25	G	-1.776328E-05	-4.529240E-06	-4.727568E-02	1.468315E-02	-9.182681E-04	0.0

SUBCASE 1

## F O R C E S O F S I N G L E - P O I N T C O N S T R A I N T

POINT ID.	TYPE	T1	T2	T3	R1	R2	R3
1	G	4.007782E+01	-4.380720E-01	2.500000E+01	0.0	0.0	0.0
6	G	-4.007782E+01	-4.380720E-01	2.500000E+01	0.0	0.0	0.0
31	G	4.007782E+01	4.380720E-01	2.500000E+01	0.0	0.0	0.0
36	G	-4.007782E+01	4.380720E-01	2.500000E+01	0.0	0.0	0.0

## S T R E S S E S I N B E A M E L E M E N T S ( C B E A M )

ELEMENT-ID	GRID	STAT DIST/ LENGTH	S T R E S S E S				S T R E S S E S		M.S.-T
			SXC	SXD	SXE	SXF	S-MAX	S-MIN	
21	31	0.000	7.455017E+01	-2.610256E+02	-2.610256E+02	7.455017E+01	7.455017E+01	-2.610256E+02	
	32	1.000	-3.252562E+02	2.461036E+01	1.387808E+02	-2.110858E+02	1.387808E+02	-3.252562E+02	
22									

## S T R E S S E S I N Q U A D R I L A T E R A L E L E M E N T S ( Q U A D 4 )

ELEMENT ID.	FIBRE DISTANCE	S T R E S S E S I N E L E M E N T C O O R D S Y S T E M			P R I N C I P A L S T R E S S E S ( Z E R O S H E A R )			VON
		NORMAL-X	NORMAL-Y	SHEAR-XY	ANGLE	MAJOR	MINOR	
1	-5.000000E-02	2.692146E+02	1.371989E+03	3.228959E+01	88.3243	1.372933E+03	2.682699E+02	1.260
	5.000000E-02	-3.365515E+02	-1.385243E+03	-1.259306E+01	-0.6879	-3.364003E+02	-1.385395E+03	1.251

■

# Пример рестарта – Метод 1

- Рестарт для вывода дополнительных результатов
  - Запрос на вывод данных по силам в элементе

```
$
$   soln5a.dat
$
RESTART
SOL 101
CEND
TITLE = STIFFENED PLATE
SUBCASE 1
    SUBTITLE = pressure load
    SPC = 1
    LOAD = 1
    FORCE = ALL
$   DISPLACEMENT=ALL
$   SPCFORCES=ALL
$   STRESS=ALL
BEGIN BULK
ENDDATA
```

# Пример рестарта – Метод 1 (Продолжение)

- Команда на запуск:

```
nastran sol5a dbs=soln5
```

- Команда, приведенная выше, указывает, что имя файла холодного старта SOLN5.DAT, а имя файла рестарта SOLN5A.DAT.

# Пример рестарта – Метод 2

- Рестарт для вывода дополнительных результатов
  - Запрос на вывод данных по силам в элементе

```
$
$   soln5b.dat
$
ASSIGN MASTER='soln5.MASTER'
RESTART
SOL 101
CEND
TITLE = STIFFENED PLATE
SUBCASE 1
    SUBTITLE = pressure load
    SPC = 1
    LOAD = 1
    FORCE = ALL
$   DISPLACEMENT=ALL
$   SPCFORCES=ALL
$   STRESS=ALL
BEGIN BULK
ENDDATA
```

# Пример рестарта – Метод 2 (Продолжение)

- Команда на запуск

```
nastran sol5b
```

- Приведенная выше команда указывает, что имя файла файла рестарта SOLN5B.DAT, а имя файла холодного старта SOLN5.DAT в данном случае прописано внутри файла рестарта с использованием оператора ASSIGN.

# Часть файла F04 (рестарт)

1

```

MACHINE                MODEL                OPERATING SYSTEM      MSC.Nastran          BUILD DATE            RUN D
Intel                   PentiumIII/550 (TINM Windows 2000 5.0 (Bu  VERSION 2001.0       OCT 12, 2000         OCT 1

```

```

=== MSC.Nastran EXECUTION SUMMARY ===

```

Day_Time	Elapsed	I/O_Mb	Del_Mb	CPU_Sec	Del_CPU	Subroutine
10:24:25	0:00	0.0	0.0	0.0	0.0	SEMTRN BGN
10:24:25	0:00	0.0	0.0	0.0	0.0	SEMTRN END
10:24:25	0:00	0.0	0.0	0.0	0.0	DBINIT BGN

```

** CURRENT PROJECT ID = ' "BLANK" ' ** CURRENT VERSION ID = 2

```

```

SUMMARY OF FILE ASSIGNMENT FOR THE PRIMARY DATABASE ( DBSNO

```

ASSIGNED PHYSICAL FILE NAME (/ORIGINAL)	LOGICAL NAME	DBSET	STATUS	BUFSIZE	CLUSTER SIZE	T
./soln5.MASTER	MASTER	MASTER	OLD	8193	1	00
./soln5.DBALL	DBALL	DBALL	OLD	8193	1	00
g:/scratch/soln5a.T1604_25.OBJSCR	OBJSCR	OBJSCR	NEW	8193	1	00
**** MEM FILE ****	* N/A *	SCRATCH				
g:/scratch/soln5a.T1604_25.SCRATCH	SCRATCH	SCRATCH	NEW	8193	1	00
g:/scratch/soln5a.T1604_25.SCR300	SCR300	SCRATCH	NEW	8193	1	00
10:24:25	0:00	0.0	0.0	0.0	0.0	DBINIT END
10:24:25	0:00	0.0	0.0	0.0	0.0	XCSA BGN

```

SUMMARY OF FILE ASSIGNMENT FOR THE DELIVERY DATABASE ( DBSNO

```

ASSIGNED PHYSICAL FILE NAME (/ORIGINAL)	LOGICAL NAME	DBSET	STATUS	BUFSIZE	CLUSTER SIZE	T
k:/nast2001/msc2001/i386/SSS.MASTERA	MASTERA	MASTER	OLD	8193	1	00
./sss.MASTERA						
k:/nast2001/msc2001/i386/SSS.MSCOBJ	MSCOBJ	MSCOBJ	OLD	8193	1	00
./sss.MSCOBJ						
k:/nast2001/msc2001/i386/SSS.MSCSOU	MSCSOU	MSCSOU	OLD	8193	1	00
./sss.MSCSOU						
10:24:26	0:01	22.0	22.0	0.2	0.2	XCSA END
10:24:26	0:01	22.0	0.0	0.2	0.0	CGPI BGN
10:24:26	0:01	22.0	0.0	0.2	0.0	CGPI END
10:24:26	0:01	22.0	0.0	0.2	0.0	LINKER BGN

# Часть файла F04 (рестарт) (Продолжение)

```
*** SYSTEM WARNING MESSAGE 6299 (PARCPU)
    NO PARALLEL PROCESS CREATED
```

```
*** TOTAL MEMORY AND DISK USAGE STATISTICS ***
```

```
+----- SPARSE SOLUTION MODULES -----+
HIWATER      SUB_DMAP      DMAP
(WORDS)    DAY_TIME    NAME      MODULE
          0  10:24:25  PREFACE  0  PREFACE
```

```
+----- MAXIMUM DISK USAGE -----+
HIWATER      SUB_DMAP      DMAP
(MB)    DAY_TIME    NAME      MODULE
      8.438  10:24:27  SESTATIC  311  EXIT
```

```
*** DATABASE USAGE STATISTICS ***
```

```
+----- LOGICAL DBSETS -----+
DBSET      ALLOCATED  BLOCKSIZE  USED  USED
           (BLOCKS)  (WORDS)   (BLOCKS)  %
MASTER          5000      8192      53   1.06
DBALL        250000      8192     100   0.04
OBJSCR         5000      8192      97   1.94
SCRATCH       500100      8192       9   0.00
```

```
+----- DBSET FILES -----+
FILE      ALLOCATED  HIWATER  HIWATER  I/O TRANSFERRE
           (BLOCKS)  (BLOCKS)  (MB)     (GE)
MASTER          5000      53      1.656    0.07
DBALL        250000     118      3.688    0.02
OBJSCR         5000      97      3.031    0.00
(MEMFILE       100      35      1.094    0.00
SCRATCH       250000       1      0.031    0.00
SCR300       250000       1      0.031    0.00
=====
TOTAL:                                     0.10
```

# Часть файла F06 (рестарт)

0 SUBCASE 1

FORCES IN BEAM ELEMENTS (CBEAM)									
ELEMENT-ID	GRID	STAT DIST/ LENGTH	- BENDING MOMENTS -		- WEB SHEARS -		AXIAL FORCE	TOTAL TORQUE	W T
			PLANE 1	PLANE 2	PLANE 1	PLANE 2			
0	21								
	31	0.000	-3.846817E+01	3.996803E-14	-1.964363E+01	-4.799914E-01	-3.543032E+01	-2.796655E-01	0.
	32	1.000	4.010637E+01	1.919966E+00	-1.964363E+01	-4.799914E-01	-3.543032E+01	-2.796655E-01	0.
0	22								
	32	0.000	2.146356E+01	1.919966E+00	-9.824739E+00	2.871786E-01	-5.304548E+01	-9.595848E-02	0.
	33	1.000	6.076252E+01	7.712511E-01	-9.824739E+00	2.871786E-01	-5.304548E+01	-9.595848E-02	0.
0	35								
	5	0.000	4.010637E+01	-1.919966E+00	1.964363E+01	-4.799914E-01	-3.543032E+01	-2.796655E-01	0.
	6	1.000	-3.846817E+01	4.618528E-14	1.964363E+01	-4.799914E-01	-3.543032E+01	-2.796655E-01	0.

# Пример рестарта - Заключение

- **Представленный выше рестарт использует ту же базу данных, что и холодный старт.**
  - Версия 1 базы данных удалялась, поскольку что не была использована опция KEEР.
  - Чтобы сохранить версию 1, необходимо задать опцию KEEР
  - Если выполняется другой рестарт, то он использует самую позднюю версию базы данных - версию 2 в данном случае.
- **В этом примере в рестарте был выполнен только процесс распечатки дополнительных результатов.**
- **ИСПОЛЬЗУЙТЕ АВТОМАТИЧЕСКИЕ РЕСТАРТЫ  
- ЭТО СОХРАНИТ ВРЕМЯ И ДЕНЬГИ!**

# Манипуляции с базой данных

- Все умолчания секции FMS могут быть изменены; однако, установки по умолчанию подходят для большинства расчетов.
- Местонахождение и максимальный размер каждого из DBSETов может быть определен пользователем.
- База данных может быть разделена на множество физических файлов, расположенных на различных дисках или на различных машинах, объединенных в сеть.

# Манипуляции с базой данных (Продолжение)

## • Операторы FMS

- ASSIGN - Назначает физическое имя файла для DBSET или FORTRAN файла, для использования другими операторами FMS или модулями DMAP.
  - INIT - Создает временный или постоянный DBSET.
  - EXPAND - Объединяет дополнительные DBSETы с существующими постоянными DBSETами. Используется при рестарте, когда предыдущий расчет заполнил выделенное пространство для DBSET.
  - DBCLEAN - удаляет из базы данных предыдущие версии.
  - DBDIR - печатает каталог базы данных.
- 
- **Полное описание операторов FMS приведено в разделе 2 MSC NASTRAN QRG.**

# Операторы ASSIGN и INIT

- **Формат (упрощенный)**

**ASSIGN логическое\_имя\_dbset = имя\_физического\_файла**

**INIT имя\_dbset LOGICAL=(логическое\_имя\_dbset(p),...)**

где

р = максимальному числу блоков отводимых для данной составляющей DBSETа – может быть определено в блоках (по умолчанию), мегабайтах (mb), мегасловах (mw), килобайтах (kb), или килословах (kw).

- **Пример:**

**INIT DBALL, logical=(DB1(1000mb),DB2(1000mb))**

- будет создан вариант DBALL, содержащий два файла, каждый из которых может вырасти до максимального размера в 1000 мегабайт
- **Размер блока машинно-зависим; для вычислений смотрите литературу: *Configuration and Operations Guide*.**

# Операторы ASSIGN и INIT (Продолжение)

- Максимальный размер используемых файлов постоянных DBSET'ов не может быть изменен при рестарте. Любой DBSET (например, DBALL) может быть расширен при рестарте добавлением используемых файлов с помощью записи EXPAND. При рестарте удалите операторы ASSIGN и INIT для уже существующих DBSETов.
- Временные DBSETы можно определять для каждого рестарта, если умолчания не адекватны.
- Оператор INIT может использоваться без оператора ASSIGN для определения размеров DBSET с именами по умолчанию.

# Примеры - ASSIGN и INIT

- Инициализация временного файла в 1000 мегабайт в директории SCR2

```
ASSIGN SCR='/scr2/test.scratch'
```

```
INIT SCRATCH,LOGICAL=(SCR(1000mb))
```

```
▪
```

```
▪
```

```
▪
```

- Инициализация DBALL с двумя DBSETами, каждый из которых имеет максимальный размер 500 мегабайт и SCRATCH с тремя DBSETами (максимальный размер каждого по 2000 мегабайт).

# Примеры - ASSIGN и INIT (Продолжение)

```
ASSIGN DB1='/home/mydir/sample.db1'  
ASSIGN DB2 ='/home/hisdir/sample.db2'  
ASSIGN SCR1 = '/scr/sample.scr1'  
ASSIGN SCR2 ='/scr/sample.scr2'  
ASSIGN SCR3 ='/scr/sample.scr3'  
$  
INIT DBALL,LOGICAL=(DB1(500mb),DB2(500mb))  
INIT SCRATCH,LOGICAL=(SCR1(2000mb),SCR2(2000mb),  
SCR3(2000mb))
```

- .
- .
- .

# Примеры - ASSIGN и INIT (Продолжение)

- **Примечание:**

- В данном случае используются маленькие буквы для имен файлов. Машины под ОС UNIX различают большие и маленькие буквы. MSC Nastran преобразует все буквы в именах файлов в большие если они не заключены в одиночные кавычки.

# Использование свободного пространства одного диска

- Позволяет избежать нескольких dbsets при наличии большого свободного пространства на одном диске
- Например, если на диске, на котором хранятся временные файлы, есть свободные 68 Гб, можно написать следующую команду

```
nastran myjob scr=65gb
```

- Эта команда отведет 65 Гб для хранения временных файлов на диске, установленном по умолчанию
- То же самое можно использовать для dball (ddbball)

# Оператор EXPAND

- Позволяет пользователю добавлять новые файлы к существующему DBSETам.
- Оператор ASSIGN должен использоваться совместно с оператором EXPAND.
- Формат:

ASSIGN логическое\_имя\_dbset= имя\_физического\_файла

EXPAND имя\_dbset LOGICAL=(логическое\_имя\_dbset(p),...)

- где p = числу блоков (или др.), отводимых DBSET

# Пример - Оператор EXPAND

- Холодный старт

```
ASSIGN DB1= '/home/nas101/sample.db1'  
ASSIGN DB2= '/home/nas101/sample.db2'  
ASSIGN SCR= '/tmp/sample.scr'  
$  
INIT DBALL,LOGICAL=(DB1(5000),DB2(5000))  
INIT SCRATCH,LOGICAL=(SCR(20000))  
$  
.  
.  
.
```

# Пример - Оператор EXPAND (Продолжение)

- Рестарт (предыдущий запуск сорвался из-за переполнения DBALL)

```
RESTART
```

```
ASSIGN MASTER='sample.MASTER'
```

```
ASSIGN DB3= '/home/nas101/sample.db3'
```

```
ASSIGN SCR= '/tmp/sample.scr'
```

```
$
```

```
EXPAND DBALL,LOGICAL=(DB3(5000))
```

```
INIT SCRATCH,LOGICAL=(SCR(20000))
```

```
$
```

```
.  
. .  
. .
```

# Пример назначения FORTRAN файла

- Следующий ALTER запишет в файл формата ASCII глобальную матрицу жесткости и матрицу масс модели, состоящей из одного элемента типа CBAR.

```

assign output4='bar.out', unit=11,form=formatted
sol 103
malter 'malter.*kgg.*mgg' $
matprn kgg,mgg// $
output4 kgg//-1/11/-1 $
output4 mgg//-2/11/-1 $
cend
title = beam model writing stiffness and mass matrices using output4
disp = all
method = 1
begin bulk
grid      1          0.      0.      0.
grid      2          1.      0.      0.
cbar      2          1          2          1.      1.      0.
pbar      1          1          .001      .001      .001      .002
mat1      1          3.E7          .3          7.32e-4
eigr1     1          12
enddata

```

# Пример назначения FORTRAN файла (продолжение)

- **BAR.OUT** результат запуска **BAR.DAT**

```

12      12      6      2KGG      1P,5E16.9
  1      1      7
3.000000142E+04 0.000000000E+00 0.000000000E+00 0.000000000E+00 0.000000000E+00
0.000000000E+00-3.000000142E+04
  2      2      11
3.600000171E+05 0.000000000E+00 0.000000000E+00 0.000000000E+00 1.800000085E+05
0.000000000E+00-3.600000171E+05 0.000000000E+00 0.000000000E+00 0.000000000E+00
1.800000085E+05
  3      3      9
3.600000171E+05 0.000000000E+00-1.800000085E+05 0.000000000E+00 0.000000000E+00
0.000000000E+00-3.600000171E+05 0.000000000E+00-1.800000085E+05
  4      4      7
2.307692310E+04 0.000000000E+00 0.000000000E+00 0.000000000E+00 0.000000000E+00
0.000000000E+00-2.307692310E+04
.
.
.
  12      2      11
1.800000085E+05 0.000000000E+00 0.000000000E+00 0.000000000E+00 6.000000285E+04
0.000000000E+00-1.800000085E+05 0.000000000E+00 0.000000000E+00 0.000000000E+00
1.200000057E+05
  13      1      1
2.057694367E-28
  12      12      6      2MGG      1P,5E16.9
  1      1      1
3.660000103E-07
  2      2      1
3.660000103E-07
  3      3      1
3.660000103E-07
  7      7      1
3.660000103E-07
  8      8      1
3.660000103E-07
  9      9      1
3.660000103E-07
  13      1      1
2.119324126E-28

```