



## Manual de Operação – USPDesigner 1.3

---

### Affine

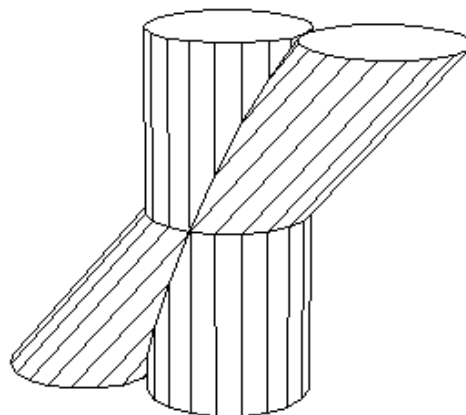
---

Função: *escala um objeto*

Forma de Chamada: **affine** <nome> <X Y Z>

Esta função escalará o objeto/grupo de nome **nome** em relação ao ponto **X Y Z**. Este comando requisitará em seguida a matriz de transformação 3x3 (cada linha requisitará 3 elementos). Esta transformação não pode ser aplicada para grupos.

```
## Transformacao afim: shap12.cad
##
camera -g cam 100 50 40 0 0 10 3 1 3 0 1 # definicao da camera
cylinder c1 20 5 20
copy c1 c2
affine c2 0 0 10
  1 0 0
  0 1 0
  0 1 1
clear -a
display cam c1 c2
```



---

### Axis

---

Função: *realiza manipulações com eixos*

Forma de Chamada: **axis** -ademrs <parâmetros>

- Define um eixo de rotação: **axis -r** <nome> <Xp Yp Zp> <Xv Yv Zv>

Define um eixo de rotação com o nome **nome**. Este eixo passa pelo ponto (**Xp, Yp, Zp**) e possui a direção (**Xv, Yv, Zv**). Este eixo será criado em associação ao grupo corrente.



## Departamento Engenharia Mecatrônica e de Sistemas Mecânicos

- Define um eixo de translação: **axis -s <nome> <Xv Yv Zv>**

Define um eixo de translação com o nome **nome**. Este eixo possui a direção (**Xv**, **Yv**, **Zv**). Este eixo será criado em associação ao grupo corrente.

- Realiza uma transformação sobre o eixo: **axis -m <nome> <N>**

Realiza uma transformação sobre o eixo de nome **nome**. Esta transformação é interpretada em concordância à natureza do eixo. Caso seja um eixo de rotação, o valor **N** é interpretado como um ângulo de rotação. Caso seja um eixo de translação o valor **N** é interpretado como um fator de deslocamento, e neste caso o deslocamento real é definido pela seguinte operação:

$$(X_d \ Y_d \ Z_d) = N \cdot (X_v \ Y_v \ Z_v) \quad ( 1 )$$

Todos os grupos, objetos e eixos associados ao eixo de nome **nome** sofrerão esta transformação.

- Associação do eixo com Grupos/Objetos: **axis -a <nome> <NomeGO>**

Associa o eixo de nome **nome** ao Grupo/Objeto de nome **NomeGO**.

- Desconecta um eixo do Grupo/Objeto: **axis -d <nome> <NomeGO>**

Desconecta o eixo de nome **nome** do Grupo/Objeto de nome **NomeGO**. O eixo é desconectado do modelador, mas continua disponível para alguma associação.

- Elimina um eixo: **axis -e <nome>**

Elimina o eixo de nome **nome**.

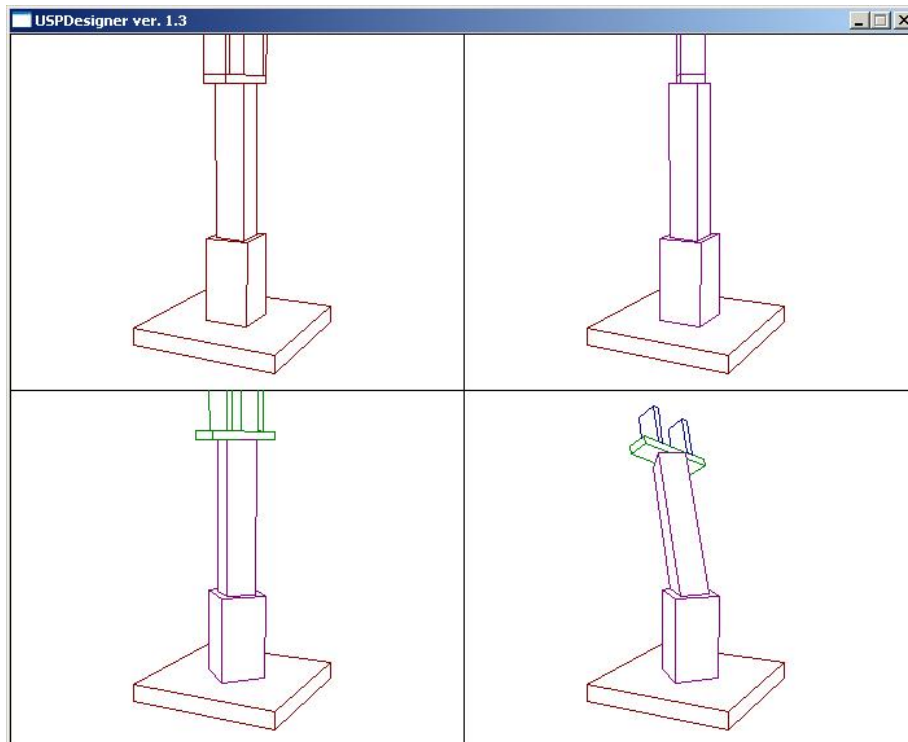
```
## robo articulado: armrobot.cad
##
clear -a
camera -g cam 80 40 30 0 0 15 3 1 3 0 0
##
## dedo esquerdo
##
group -G left
cube lf 1 3 5
move lf 0 -1.5 0
group -c \
##
## dedo direito
##
group -G right
cube rf 1 3 5
move rf -1 -1.5 0
group -c \
##
## garra
##
group -G hand
group -p hand left

group -p hand right
cube d 12 3 1
move d -6 -1.5 -1
axis -s ml -5 0 0
axis -a ml left
axis -s mr 5 0 0
axis -a mr right
axis -m ml -1
axis -m mr -1
move hand 0 0 1
##
## primeira parte do braco
##
group -c \
group -G gc
group -p gc hand
cube c 4 4 20
move c -2 -2 0
move hand 0 0 20
axis -r m3 0 0 20 0 0 1
axis -a m3 hand
##
## segunda parte do braco
```



```
##
group -c \
group -G gb
group -p gb gc
cube b 6 6 10
move b -3 -3 0
move gc 0 0 8
axis -r m2 0 0 9 1 0 0
axis -a m2 gc
## abaixar o braco em 90 graus
axis -m m2 -90
##
## robo completo
##
group -c \
group -G robot
group -p robot gb
cube a 20 20 2
move a -10 -10 0
move gb 0 0 2
axis -r m1 0 0 2 0 0 1
axis -a m1 gb
##
clear -g
##

## demonstrativo
##
wind 0 0.5 0.5
axis -m m2 90 #30
color -g robot 4
disp cam robot
##
wind 0.5 0.5 0.5
axis -m m3 30
color -g gb 5
disp cam robot
##
wind 0 0 0.5
axis -m m1 -45
axis -m m1 0.5
axis -m mr 0.5
color -g hand 2
disp cam robot
##
wind 0.5 0 0.5
axis -m m2 30
color -g left 1
color -g right 1
disp cam robot
```





## Camera

---

Função: *criação e edição de câmeras*

Forma de Chamada: **camera** -defghmrs <parâmetros>

Criação de câmeras: **camera** -g <n><Ex Ey Ez><Rx Ry Rz><s><m><e><h><a>

Ao se definir uma câmera é necessário que todos os parâmetros sejam fornecidos. Caso a câmera já exista, a câmera será redefinida com os parâmetros fornecidos.

- **n**: é o nome da câmera, que possui no máximo 12 letras. Operações que manipulem os dados desta câmera devem referenciá-la pelo seu nome.
- **E**: é a posição em coordenadas tri-dimensionais da câmera. Este parâmetro é o conjunto de três números reais representando as coordenadas **x**, **y** e **z**.
- **R** (ponto de referência): é a posição em coordenadas tri-dimensionais que define o ponto central de visão. O olho e o ponto de referência definem uma direção principal de visão.
- **s** (abertura): este parâmetro define a abertura da câmera. Se possuir o valor 1.0 o ângulo de visão será de 90 graus. Quanto maior for o seu valor, maior será o ângulo de visão; e quanto menor for o seu valor, menor será o ângulo de visão.
- **m**: este parâmetro define o modo de projeção da câmera:
  - **m = 1** - projeção perspectiva
  - **m = 2** - projeção paralela
  - **m = 3** - projeção em três vistas
- **e**: este parâmetro define o processamento das arestas escondidas:
  - **e = 0** - não ocorre nenhum processamento
  - **e = 1** - processamento local
  - **e = 2** - processamento global
  - **e = 3** - processamento global e cálculo das intersecções entre objetos.

O processamento local realiza o processamento das arestas de um objeto que são escondidas pelo próprio objeto. O processamento global realiza o processamento das arestas que são escondidas em relação a todos os objetos da cena.

- **h**: este parâmetro determina se as arestas invisíveis são desenhadas ou não. As arestas invisíveis são desenhadas com linhas tracejadas.
  - **h = 0** - não são desenhadas
  - **h = 1** - são desenhadas
- **a**: este parâmetro determina se as arestas que separam duas faces quase paralelas devem ser desenhadas ou não.
  - **a = 0** - não são desenhadas
  - **a = 1** - são desenhadas



---

## Departamento Engenharia Mecatrônica e de Sistemas Mecânicos

- Alteração do olho: **camera -e <nome> <Ex Ey Ez>**  
O **nome** deve corresponder a uma câmera já existente. O olho da câmera deve representar a nova posição da câmera no espaço tri-dimensional.
- Alteração do ponto de referência: **camera -r <nome> <Rx Ry Rz>**  
O **nome** deve corresponder a uma câmera já existente. O **R** deve representar um novo ponto de referência no espaço tri-dimensional.
- Alteração do abertura: **camera -s <nome> <s>**  
O **nome** deve corresponder a uma câmera já existente. O **s** deve representar o novo ângulo de visão da câmera.
- Alteração do modo: **camera -m <nome> <m>**  
O **nome** deve corresponder a uma câmera já existente. O **m** deve representar o novo de projeção da câmera.
- Alteração do elim: **camera -h <nome> <e>**  
O **nome** deve corresponder a uma câmera já existente. O **e** deve representar o novo processamento de arestas escondidas da câmera.
- Alteração do hash: **camera -d <nome> <h>**  
O **nome** deve corresponder a uma câmera já existente. O **h** deve indicar se as arestas invisíveis devem ser desenhadas ou não.
- Alteração do arest: **camera -f <nome> <a>**  
O **nome** deve corresponder a uma câmera já existente. O **a** deve indicar se as arestas que separam duas faces quase paralelas devem ser desenhadas ou não.

```
## exemplos de camera: camera.cad
# exibe todas as linhas
camera -g cam0      100 50 40 0 0 0 4 1 0 0 0
# processamento local
camera -g caml1 100 50 40 0 0 0 4 1 1 1 1 # linhas hash., exibe planos
camera -g caml2 100 50 40 0 0 0 4 1 1 1 0 # linhas hash., nao exibe planos
camera -g caml3 100 50 40 0 0 0 4 1 1 0 1 # nao linhas hash., exibe planos
camera -g caml4 100 50 40 0 0 0 4 1 1 0 0 # nao linhas hash., nao exibe planos
## processamento global
camera -g camh1 100 50 40 0 0 0 4 1 2 1 1 # linhas hash., exibe planos
camera -g camh2 100 50 40 0 0 0 4 1 2 1 0 # linhas hash., nao exibe planos
camera -g camh3 100 50 40 0 0 0 4 1 2 0 1 # nao linhas hash., exibe planos
camera -g camh4 100 50 40 0 0 0 4 1 2 0 0 # nao linhas hash., nao exibe planos
## processamento completo
camera -g cami1 100 50 40 0 0 0 4 1 3 1 1 # linhas hash., exibe planos
camera -g cami2 100 50 40 0 0 0 4 1 3 1 0 # linhas hash., nao exibe planos
camera -g cami3 100 50 40 0 0 0 4 1 3 0 1 # nao linhas hash., exibe planos
camera -g cami4 100 50 40 0 0 0 4 1 3 0 0 # nao linhas hash., nao exibe planos
```



## Clear

---

Função: *apagar a tela*

Forma de Chamada: **clear -acgw**

- Apaga a tela gráfica e a tela alfanumérica: **clear -a**
- Apaga a janela atual: **clear -w**

## Color

---

Função: *manipula a cor dos objetos/grupos*

Forma de Chamada: **color -dfgo <nome> <cor>**

Esta função permite definir a cor de objetos/grupos. Para que a nova cor seja visualizada na tela é necessário executar novamente o comando **DISPLAY**. O sistema possui o seguinte código de cores:

| Código | Cor          |
|--------|--------------|
| -1     | valor neutro |
| 0      | preto        |
| 1      | vermelho     |
| 2      | verde        |
| 3      | azul         |
| 4      | azul claro   |
| 5      | amarelo      |
| 6      | violeta      |
| 7      | branco       |

- Define a cor de um objeto: **color -o <nome> <cor>**

A cor do objeto de nome **nome** é definida por **cor**. Caso o valor de cor seja **-1**, este objeto assumirá a cor de algum grupo ancestral.

- Define a cor de um grupo: **color -g <nome> <cor>**

A cor do grupo de nome **nome** é definida por **cor**. Caso o valor de cor seja **-1**, este grupo assumirá a cor de algum grupo ancestral.

- Define a cor do grupo raiz: **color -d <cor>**

O grupo raiz possui a cor inicial branca. Caso o grupo raiz seja definido com os valores **-1** ou **0**, é possível que alguns objetos fiquem invisíveis.

- Define a cor da janela: **color -f <cor>**



A janela de visualização será exibida com a cor definida por **cor**. Não pode ser utilizado o valor **-1** para esta função.

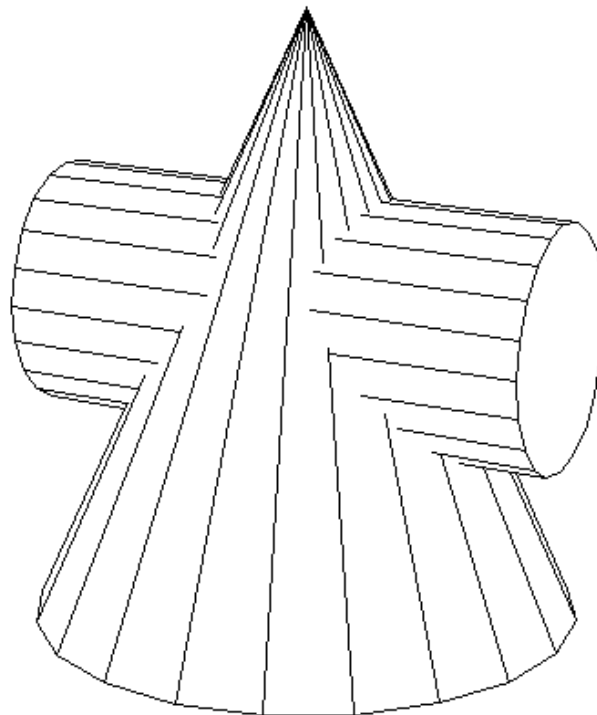
## Cone

Função: *definir um cone sólido*

Forma de Chamada: **cone** <nome> <n> <R> <H>

Esta função define um cone de nome **nome** pela aproximação de sua base por um polígono de **n** lados. O cone possui raio **R** e altura **H**. O centro da base do cone é posicionado na origem dos eixos **XYZ**, e o cone é criado na direção positiva do eixo **Z**. O valor de **n** deve ser maior que 2, o valor de **R** deve ser positivo e **H** deve ser positivo.

```
# Cone e cilindro: shapel.cad
camera -g cam1 100 50 40 0 0 10 5 1 2 0 1 # definicao da camera
cone a 20 10 20 # criacao do cone
cylinder b 20 4 20 # criacao do cilindro
rotate b 0 0 0 1 0 0 90 # rotacao do cilindro
move b 0 10 10 # translacao do cilindro
clear -a # apaga a area grafica
disp cam1 * # exhibe os elementos
```





## Copy

---

Função: *copiar um objeto ou grupo*

Forma de Chamada: `copy <nomeO> <nomeC>`

Esta função copia o objeto/grupo de nome **nomeO** em um novo objeto/grupo de nome **nomeC**. O novo objeto/grupo são definidos como filho do grupo corrente. Caso algum elemento de nome **nomeC** já esteja definido o sistema enviará uma mensagem de erro. Caso esteja-se copiando um grupo, além do grupo especificado por **nomeO**; serão copiados todos os objetos que forem filhos deste grupo. Estes objetos que forem copiados, possuirão os mesmos nomes que os objetos originais.

## Cube

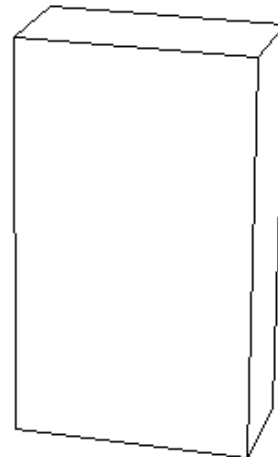
---

Função: *definir um cubo sólido*

Forma de Chamada: `cube <nome> <A> <B> <C>`

Esta função define um cubo de nome **nome** cujo comprimento dos lados são dados por **A**, **B** e **C**. O cubo é criado no semi-espaco positivo definido pelo eixos **XYZ**. Um dos vértices do cubo está sobre a origem dos eixos, o cubo possui o comprimento **A** sobre o eixo **X**, comprimento **B** sobre o eixo **Y** e comprimento **C** sobre o eixo **Z**. As dimensões **A**, **B** e **C** devem ser não negativas. Até duas dimensões podem ser nulas simultaneamente.

```
# cubos
camera -g cam1 100 30 40 0 0 0 4 1 3 0 0
cube a 10 10 10
clear -a
display cam1 a
wait 10
cube b 20 20 1
clear -a
display cam1 b
wait 10
cube c 5 10 15
clear -a
display cam1 c
```







## Cylinder

---

Função: *definir um cilindro sólido*

Forma de Chamada: **cylinder** <nome> <n> <R> <H>

Esta função define um cilindro de nome **nome** pela aproximação de sua base por um polígono de **n** lados. O cilindro possui raio **R** e altura **H**. O centro da base do cilindro é posicionado na origem dos eixos **XYZ**, e o cilindro é criado na direção positiva do eixo **Z**. O valor de **n** deve ser maior que 2, o valor de **R** deve ser positivo e **H** deve ser não negativo.

## Delete

---

Função: *apaga um objeto*

Forma de Chamada: **delete** <nome>

Esta função apaga o objeto de nome **nome** que deve estar presente no grupo corrente. Substituindo-se o nome do objeto por **\***, apaga-se todos os objetos filhos do grupo corrente.

## Dir

---

Função: *fornece o diretório*

Forma de Chamada: **dir** <parametros>

Este comando deve ser utilizado da mesma forma que o comando **DIR** do MS-DOS. É possível fornecer a extensão dos arquivos, máscara de nome e diretório, entre outras opções.

## Dispcoor

---

Função: *exibe na tela o eixo de coordenadas*

Forma de Chamada: **dispcoor** -am <parâmetros>

Esta função exibe na tela um eixo de coordenadas e/ou um plano quadriculado paralelo a um dos eixos de coordenadas. Para fazer uso desta função é necessário que uma câmera monitora esteja definida. Esta função objetiva auxiliar o posicionamento de objetos.

- Exibe eixo de coordenadas: **dispcoor -a <comprimento> <cor>**  
Exibe o eixo de coordenadas **XYZ** com o comprimento definido por **comprimento** e com a cor definida por **cor**.
- Exibe o plano **YZ** quadriculado: **dispcoor -m x=<v> <c> <esp> <c1> <c2>**



---

## Departamento Engenharia Mecatrônica e de Sistemas Mecânicos

Exibe o plano  $x=<v>$  quadriculado, a espessura do reticulado é definida por **esp** e a cor do reticulado é definida por **c2**. O eixo de coordenadas é exibido com o comprimento definido por **c** e a cor definida por **c1**.

- Exibe o plano **XZ** quadriculado: **dispcoor -m y=<v> <c> <esp> <c1> <c2>**

Exibe o plano  $y=<v>$  quadriculado, a espessura do reticulado é definida por **esp** e a cor do reticulado é definida por **c2**. O eixo de coordenadas é exibido com o comprimento definido por **c** e a cor definida por **c1**.

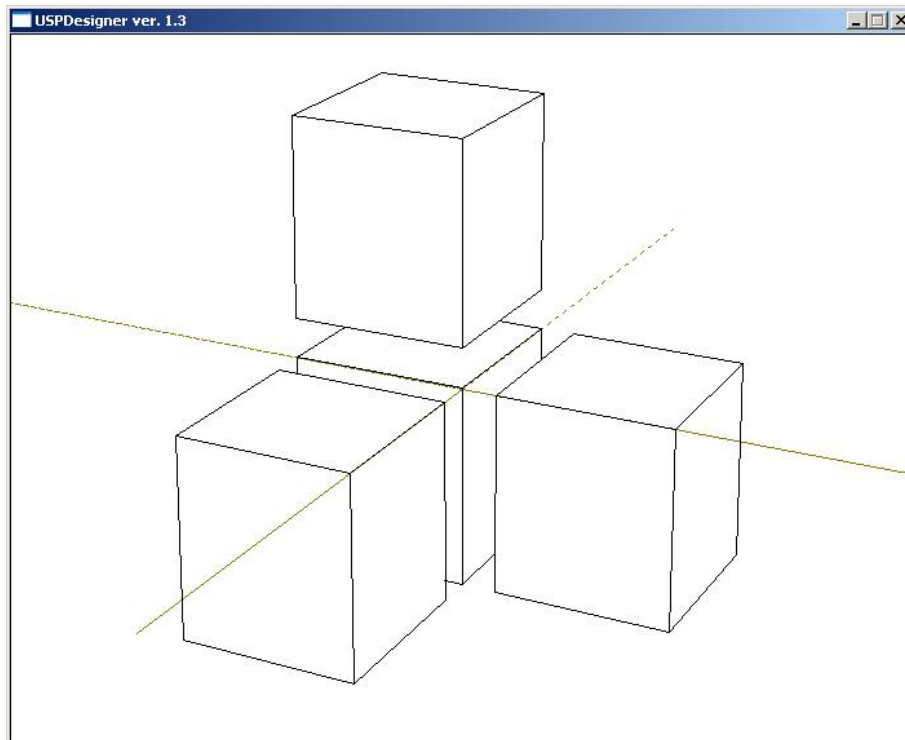
- Exibe o plano **XY** quadriculado: **dispcoor -m z=<v> <c> <esp> <c1> <c2>**

Exibe o plano  $z=<v>$  quadriculado, a espessura do reticulado é definida por **esp** e a cor do reticulado é definida por **c2**. O eixo de coordenadas é exibido com o comprimento definido por **c** e a cor definida por **c1**.

```
## coordemo
clear -a
camera -g cam1 100 50 40 0 0 0 5 1 1 1 1
camera -g cam2 100 50 40 0 0 0 5 1 3 1 0
camera -g cam3 100 50 40 0 0 0 5 1 3 0 0
camera -g cam4 100 50 40 0 0 0 5 3 3 0 1
monitor cam1
monitor off
##
## criacao do paralelepipedo a
##
clear -g
print Criaremos um cubo com todos os lados iguais a 10.\n
dispcoor -m z=0 30 5 6 3
dispcoor -a 30 6
cube a 10 10 10
disp cam1 a
wait 2
##
move a -10 -10 -10
clear -a
dispcoor -a 30 3
disp cam1 a
wait 2
##
## cubos b c d sao copias do cubo a, deslocamos os cubos de 12 nas coordenadas
##                                     x,y,z
##
copy a b
move b 12 0 0
disp cam1 b
copy a c
move c 0 12 0
disp cam1 c
copy a d
move d 0 0 12
disp cam1 d
wait 2
##
```



```
## exibe os elementos com processamento completo
##
clear -g
dispcoor -m z=0 30 5 6 3
disp cam3 a b c d
wait 2
##
## desloca-se o cubo a de (6,6,6)
##
move a 6 6 6
clear -g
dispcoor -m z=0 30 5 6 3
disp cam3 a b c d # projecao axonometrica
wait 2
clear -g
disp cam4 a b c d # tres vistas
wait 2
##
## faz com que o cubo a passe pela origem (1,1,1) e rotaciona-o de 30 graus
##
rotate a 0 0 0 1 1 1 30
clear -g
dispcoor -m z=0 30 5 6 3
disp cam3 a b c d # projecao axonometrica
wait 2
clear -g
disp cam4 a b c d # tres vistas
```





## Dispfile

---

Função: *exibe/escreve arquivo de visualização*

Forma de Chamada: **dispfile -rw <arquivo>**

Ao se acionar esta função, a imagem que for exibida na tela pelo comando **DISPLAY** será armazenada em um arquivo definido por **arquivo**. Da mesma maneira, esta imagem poderá ser lida e exibida na tela.

- Criação do arquivo imagem: **dispfile -w <arquivo>**

A imagem que for exibida pelo próximo comando **DISPLAY** será armazenada no arquivo de nome **arquivo**. Quando o comando **DISPLAY** for encerrado o arquivo imagem é fechado.

- Leitura do arquivo imagem: **dispfile -r <arquivo>**

A imagem armazenada no arquivo imagem será exibida na window corrente. O arquivo imagem é escrito em caracteres ASCII, de maneira que é possível observá-lo por um editor de textos. Cada linha do arquivo corresponde à definição de uma linha gráfica bi-dimensional. Cada linha do arquivo do arquivo possui a seguinte estrutura:

| Posição        | conteúdo              | faixa        |
|----------------|-----------------------|--------------|
| primeiro valor | valor da cor da linha | (0-7)        |
| segundo valor  | tipo de linha         | (1-2)        |
| terceiro valor | coordenada x inicial  | (-1.0 - 1.0) |
| quarto valor   | coordenada y inicial  | (-0.7 - 0.7) |
| quinto valor   | coordenada x final    | (-1.0 - 1.0) |
| sexto valor    | coordenada y final    | (-0.7 - 0.7) |

```
## criacao do anel - comando dispfile -w ringsn.dsp
##
camera -g cam0      100 50 40 0 0 0 4 1 0 0 0
camera -g camh3 100 50 40 0 0 0 4 1 2 0 1
camera -g cami3 100 50 40 0 0 0 4 1 3 0 1
##
torus r1 10 15 5
copy r1 r2
rotate r2 0 0 0 1 0 0 90
dispfile -w rings1.dsp
disp cami3 r1 r2
##
rotate r2 0 0 0 0 0 1 90
move r1 0 -7.5 0
move r2 0 7.5 0
cl -a
dispfile -w rings2.dsp
disp camh3 r1 r2
```



---

Departamento Engenharia Mecatrônica e de Sistemas Mecânicos

## Display

---

Função: *exibir na tela um conjunto de objetos por uma câmera*

Forma de Chamada: **display** <camera> <lista>

A câmera deve ser definida pelo seu nome, caso a câmera não exista este comando não terá nenhum efeito. O parâmetro <lista> define o conjunto de objetos/grupos que devem ser exibidos na tela. Cada objeto/grupo é definido por seu nome. O parâmetro <lista> pode ser representado pelo caracter \*, deste modo o grupo atual será exibido na tela pela câmera especificada. O parâmetro <lista> também pode ser representado pelo caracter \*\*, deste modo todos os objetos/grupos serão exibidos na tela pela câmera especificada.

Este comando exibirá os elementos definidos sobre a tela atual. Caso seja necessário apagar a tela atual, faça uso do comando **Clear**.

## End

---

Função: *termina o sistema*

Forma de Chamada: **end**

Este comando termina o sistema.

## File

---

Função: *processamento batch*

Forma de Chamada: **file** <nome>

Este comando executará em modo batch o arquivo de nome **nome**. Caso a terminação do arquivo não seja especificada, será assumida a terminação **\*.CAD**. O arquivo deve estar no formato ASCII, e deve ser criado por meio de um editor de textos. Caso o arquivo referencie o comando **END**, o processamento será interrompido e o sistema retornará ao estado de entrada de dados pelo teclado. O arquivo pode referenciar outro arquivo pelo comando **FILE**. Neste caso o processamento do arquivo inicial será encerrado e o processamento do arquivo referenciado será iniciado. Ao se iniciar o sistema, o processamento do arquivo **AUTODEMO.CAD**, caso exista, será automaticamente iniciado.

## Group

---

Função: *operações sobre grupos*

Forma de Chamada: **group** -cdegGmp <parametros>



---

Departamento Engenharia Mecatrônica e de Sistemas Mecânicos

Esta função permite realizar operações sobre grupos. Os grupos estão estruturados em forma de árvore, aonde possuímos o grupo pai e o grupo filho. Possuímos dois conceitos importantes: grupo raiz e grupo corrente. O grupo raiz é o pai de todos os grupos e o grupo corrente define o grupo pai dos objetos que estão sendo definidos neste momento.

- Exibir o nome do grupo corrente: **group -c**
- Fazer com que a raiz seja o grupo corrente: **group -c \**
- Fazer com que o grupo corrente seja o pai do atual grupo corrente: **group -c ..**
- Determinar o grupo corrente: **group -c <nome>**

O grupo corrente é definido pelo grupo de nome **nome**.

- Apagar o sólido que pertence a um grupo específico: **group -d <Gnome> <Snome>**  
Apaga o sólido de nome **Snome** que pertence ao grupo de nome **Gnome**.

- Apagar todos os sólidos de um grupo específico: **group -e <Gnome>**  
Apaga todos os sólidos que possuem o grupo **Gnome** como pai.

- Cria um grupo filho: **group -g <Gnome>**  
Cria o grupo de nome **Gnome** como sendo filho do grupo corrente.

- Cria um grupo filho e o torna grupo corrente: **group -G <Gnome>**  
Cria o grupo de nome **Gnome** como sendo filho do grupo corrente e o torna grupo corrente.

- Transfere sólido: **group -m <Gnome> <Snome>**  
Transfere o sólido de nome **Snome** que é filho do grupo corrente para o grupo de nome \$Gnome\$.

- Transfere grupo: **group -p <Gpname> <Gfname>**  
Transfere o grupo de nome **Gfname** e todos os seus descendentes para o grupo de nome **Gpname**. O grupo de nome **Gpname** torna-se o grupo corrente.

## Help

---

Função: *função de auxílio*

Forma de Chamada: **help**

Todos os possíveis comandos serão listados na tela.

## Intesect

---

Função: *determinar a intersecção entre dois sólidos*



---

Departamento Engenharia Mecatrônica e de Sistemas Mecânicos

Forma de Chamada: **inteselect** <nome1> <nome2> <nome3>

Esta função determina o sólido resultado da intersecção entre os sólidos de nomes **nome1** e **nome2**. O resultado é associado ao sólido de **nome3**.

## List

---

Função: *analisar os elementos do sistema*

Forma de Chamada: **list -acfgo** <parametro>

- Volume de elementos: **list -f**

Este comando permite analisar o número de elementos que foram modelados até o momento. Possuímos informações sobre os seguintes elementos:

|                    |
|--------------------|
| número de câmeras  |
| número de eixos    |
| número de grupos   |
| número de vértices |
| número de arestas  |
| número de faces    |
| número de objetos  |

- Objetos: **list -o**

Este comando permite analisar os objetos que pertencem ao grupo corrente.

- Eixos: **list -a**

Este comando exibe os nomes de todos os eixos.

- Câmera: **list -c**

Este comando exibe os nomes de todas as câmeras.

- Grupo: **list -g**

Este comando exibe os nomes de todos os grupos.

- Relacionamento entre Grupos: **list -g <nome>**

Este comando exibe o grupo ancestral e todos os grupos descendentes do grupo de nome **nome**.

## Load

---

Função: *recupera um sólido que está salvo em arquivo*

Forma de Chamada: **load** <nome1> <nome2>



---

Departamento Engenharia Mecatrônica e de Sistemas Mecânicos

Esta função recupera um sólido que está salvo no arquivo de nome **nome2**, o sólido recuperado é associado ao nome **nome1**.

### Minus

---

Função: *determinar a diferença entre dois sólidos*

Forma de Chamada: **minus** <nome1> <nome2> <nome3>

Esta função determina o sólido resultado da diferença entre os sólidos de nome **nome1** e **nome2**. O resultado é associado ao sólido de nome **nome3**.

### Monitor

---

Função: *define uma câmera monitora*

Forma de Chamada: **monitor** <parametros>

Esta função define uma câmera monitora, que permitirá verificar a criação e edição dos objetos. Pode ser definida apenas uma câmera monitora por vez. Caso a câmera monitora esteja presente o resultado do acionamento das seguintes funções será exibido na tela:

|          |
|----------|
| Cube     |
| Cylinder |
| Cone     |
| Prism    |
| Pyramid  |
| Sphere   |
| Torus    |
| Revolve  |
| Input    |
| Move     |
| Rotate   |
| Affine   |
| Axis     |

- Definição da câmera monitora: **monitor** <camera>

A câmera monitora é definida pelos parâmetros da câmera **camera**. Neste instante a câmera monitora é ligada.

- Ligar a câmera monitora: **monitor ON**

Caso a câmera monitora já estiver definida, esta função a liga.





---

Departamento Engenharia Mecatrônica e de Sistemas Mecânicos

- Desliga a câmera monitora: **monitor OFF**

Caso a câmera monitora já estiver definida, esta função a desliga.

OBS: É necessário que uma câmera monitora esteja definida para ativar a função **DISPCOOR**. A função **DISPCOOR** deverá funcionar mesmo que a câmera monitora esteja desligada.

### Move

---

Função: *translaciona um objeto/grupo*

Forma de Chamada: **move <nome> <X Y Z>**

Esta função deslocará o objeto/grupo de nome **nome** relativamente pelo valor definido pelo vetor **x y z**. Ao realizar a translação de um grupo, todas as entidades associadas a este grupo e aos seus descendentes sofrerão a mesma translação: eixos, grupos filhos e objetos filhos.

### Ncopy

---

Função: *copiar um objeto em vários outros objetos*

Forma de Chamada: **ncopy -ls <parametros>**

Esta função copia um objeto transformando-o para outra posição.

- Copia translacionando o objeto original: **ncopy -l <nm> <Xi Yi Zi> <Xd Yd Zd> <n>**  
O objeto especificado por **nm** será copiado **n** vezes. O primeiro objeto copiado deverá ser posicionado na posição determinada pelas coordenadas **Xi Yi Zi**, e os outros objetos serão posicionados por translações aditivas no valor de **Xd Yd Zd**. Os nomes dos novos objetos serão criados automaticamente pelo sistema. Os objetos serão criados como filhos do grupo corrente.

- Copia rodando e translacionando o objeto original:

**ncopy -s <n> <Xp Yp Zp> <Xv Yv Zv> <ang> <D> <n>**

O objeto especificado por nome será copiado **n** vezes. O objeto a ser copiado será rotacionado em torno do eixo definido pelo ponto **Xp Yp Zp** e pela direção **Xv Yv Zv**. O primeiro objeto copiado será posicionado sobre o objeto original. Os objetos seguintes sofrerão uma rotação e uma translação adicionais de um ângulo definido por **ang** e um deslocamento sobre o eixo de rotação definido por **D**. Os nomes dos novos objetos serão criados automaticamente pelo sistema. Os objetos serão criados como filhos do grupo corrente.

### New

---

Função: *reinicia o sistema*

Forma de Chamada: **new**



---

Departamento Engenharia Mecatrônica e de Sistemas Mecânicos

Este comando reinicia o sistema. Caso este comando seja referenciado em um arquivo de comandos, o sistema será reiniciado e a execução do arquivo de comandos continuará normalmente.

## Print

---

Função: *imprime caracteres*

Forma de Chamada: **print** <cadeia de caracteres>

Este comando imprime uma cadeia de caracteres na tela. Caso esteja se utilizando do dispositivo ANSI, qualquer comando aceito pela norma pode ser utilizado. Os seguintes caracteres especiais podem ser utilizados:

|                   |                                                 |
|-------------------|-------------------------------------------------|
| <code>\n</code>   | passa-se a nova linha (newline)                 |
| <code>\xhh</code> | <b>hh</b> deve conter o código da letra em hexa |
| <code>^c</code>   | caracter de controle                            |
| <code>\\</code>   | representa o caracter \                         |
| <code>\^</code>   | representa o caracter ^                         |

## Prism

---

Função: *definir um prisma sólido*

Forma de Chamada: **prism** <nome> <n> <H>

Esta função define um prisma reto de nome **nome** cuja base é um polígono de **n** lados posicionada sobre o plano **xy**. O prisma reto possui altura **H**. Os vértices do polígono da base são definidos por seus pares de coordenadas no plano **xy**. O prisma é criado na direção positiva do eixo **z**. Para definir o polígono da base o usuário deve fornecer um par de valores de coordenadas por linha e em sentido anti-horário. O valor das coordenadas de cada vértice definirá a sua posição sobre o plano **xy**. O valor de **n** deve ser maior que 2, o valor de **R** deve ser positivo e **H** deve ser não negativo.

## Pyramid

---

Função: *definir uma pirâmide sólida*

Forma de Chamada: **pyramid** <nome> <n> <H>

Esta função define uma pirâmide de nome **nome** cuja base é um polígono de **n** lados posicionada sobre o plano **xy**. A pirâmide possui altura **H**. Os vértices do polígono da base são definidos por seus pares de coordenadas no plano **xy**. A altura da pirâmide é definida sobre o eixo **z** em seu sentido positivo. Para definir o polígono da base o usuário deve fornecer um par de valores de coordenadas por linha e em sentido anti-horário. O valor das coordenadas de cada vértice definirá a



---

Departamento Engenharia Mecatrônica e de Sistemas Mecânicos

sua posição sobre o plano **XY**. O valor de **n** deve ser maior que 2, o valor de **R** deve ser positivo e **H** deve ser positivo.

### **Rename**

---

Função: *modifica o nome de uma entidade*

Forma de Chamada: **rename** <Vnome> <Nnome>

Esta função permite alterar o nome de várias entidades: câmeras, grupos, eixos e objetos. A entidade de nome **Vnome** possuirá o seu nome alterado para **Nnome**.

### **Revolve**

---

Função: *definir um sólido de revolução*

Forma de Chamada: **revolve** <nome> <modo> <n> <N>

Esta função define um sólido de revolução de nome **nome** cuja revolução será aproximada por um polígono de **n** lados. A superfície a ser revolucionada é definida por polígono de **N** lados posicionado sobre o plano **RZ** (coordenadas cilíndricas). O centro do eixo de revolução é o eixo **Z**. Para definir o polígono que deve ser revolucionado o usuário deve fornecer um par de valores de coordenadas por linha e em sentido horário. O valor das coordenadas de cada vértice definirá a sua posição sobre o plano **RZ**. Existem dois modo de revolução:

- **modo0** - o sólido de revolução está afastado do eixo de rotação.
- **modo1** - o sólido de revolução está não está afastado do eixo de rotação. Neste caso o raio **R0 = RN = 0**.

O valor de **n** deve ser maior que 2 e o valor de **N** também deve ser maior que 2.

### **Rotate**

---

Função: *rotaciona um objeto/grupo*

Forma de Chamada: **rotate** <nome> <Xe Ye Ze> <Xv Yv Zv> <ang>

Esta função rotacionará o objeto/grupo de nome **nome** em torno do eixo definido pelo ponto **Xe Ye Ze** e pelo vetor **Xv Yv Zv**, de um ângulo definido por **ang**. Ao realizar a rotação de um grupo, todas as entidades associadas a este grupo e aos seus descendentes sofrerão a mesma rotação: eixos, grupos filhos e objetos filhos. Ao se realizar a rotação assume-se que o eixo está direcionado pelo sentido do vetor, e rotaciona-se de um ângulo **ang** no sentido horário.



---

## Save

Função: *salva um sólido em arquivo*

Forma de Chamada: **save** <nome1> <nome2>

Esta função salva um sólido de nome **nome1** no arquivo de nome **nome2**.

---

## Show

Função: *exibe informações sobre o modelo*

Forma de Chamada: **show** -acdefgov <parametros>

- Exibe eixos: **show -a**

Exibe todos os eixos presentes no modelo e seus parâmetros.

- Exibe eixo específico: **show -a <nome>**

Exibe os parâmetros do eixo de nome **nome**.

- Exibe câmeras: **show -c**

Exibe todas as câmeras presentes no modelo e seus parâmetros.

- Exibe câmera específica: **show -c <nome>**

Exibe os parâmetros da câmera de nome **nome**.

- Exibe grupos: **show -g**

Exibe todos os grupos presentes no modelo e seus parâmetros.

- Exibe grupo específico: **show -g <nome>**

Exibe os parâmetros do grupo de nome **nome**.

- Exibe objetos: **show -o**

Exibe todos os objetos presentes no modelo e seus parâmetros.

- Exibe objeto específico: **show -o <nome>**

Exibe os parâmetros do objeto de nome **nome**. Caso se deseje utilizar as opções **-vef** da função **show**, é necessário utilizar-se desta opção para estabelecer a que objeto estamos referenciando.

- Exibe vértice: **show -v**

Exibe os vértices do objeto especificado anteriormente.

- Exibe vértice específico: **show -v <N>**

Exibe o vértice de número **N** relativo ao objeto especificado anteriormente.



---

Departamento Engenharia Mecatrônica e de Sistemas Mecânicos

- Exibe arestas: **show -e**  
Exibe as arestas do objeto especificado anteriormente.
- Exibe aresta específica: **show -e <N>**  
Exibe a aresta de número **N** relativa ao objeto especificado anteriormente.
- Exibe faces: **show -f**  
Exibe as faces do objeto especificado anteriormente.
- Exibe face específica: **show -f <N>**  
Exibe a face de número **N** relativa ao objeto especificado anteriormente.

## Sphere

---

Função: *definir uma esfera sólida*

Forma de Chamada: **sphere** <nome> <n> <R>

Esta função define uma esfera de nome **nome** cujo círculo de base será aproximado por um polígono de **n** lados. O centro da esfera é a origem dos eixos **XYZ**. O valor de **n** deve ser maior que 1 e o valor de **R** deve ser positivo.

## Split

---

Função: *determinar o corte de um sólido*

Forma de Chamada: **split** <nome1> <nome2> <nome3> <a> <b> <c> <d>

Esta função determina o corte do sólido de nome **nome1** por um plano de equação **ax+by+cz+d=0**. A equação do plano divide o espaço tridimensional em dois semi-espacos: positivo e negativo. O sólido **nome2** corresponde à parte do sólido **nome1** que está sob o espaço positivo. O sólido **nome3** corresponde à parte do sólido **nome1** que está sob o espaço negativo.

## Time

---

Função: *fornece a hora corrente*

Forma de Chamada: **time**

Este comando fornece a hora corrente. O sistema requisita a hora corrente do sistema; portanto, caso ela esteja incorreta o relógio do sistema deve ser corrigido.



## Torus

---

Função: *definir um toróide sólido*

Forma de Chamada: **torus** <nome> <n> <R> <r>

Esta função define um toróide de nome **nome** cujos diâmetros serão aproximados por polígonos de **n** lados. O centro do toróide é a origem dos eixos **XYZ**. O raio maior é definido por **R** e o raio menor é definido por **r**. O valor de **n** deve ser maior que 2 e o valor de **R** deve ser positivo.

## Union

---

Função: *determina a união entre dois sólidos*

Forma de Chamada: **union** <nome1> <nome2> <nome3>

Esta função determina o sólido resultado da união entre os sólidos de nomes **nome1** e **nome2**. O resultado é associado ao sólido de nome **nome3**.

## Wait

---

Função: *aguarda um intervalo de tempo*

Forma de Chamada: **wait** <N>

Este comando faz com que o sistema aguarde **N** segundos para executar o próximo comando. O comando **WAIT** geralmente é referenciado em arquivos de comandos.

## Window

---

Função: *definir a viewport*

Forma de Chamada: **window** <X> <Y> <W>

Esta função define a parte da tela que será utilizada para exibir os objetos/grupos por uma câmera específica. As coordenadas **X** e **Y** definem o canto inferior esquerdo da viewport, as coordenadas **X+W** e **Y+W** definem o canto superior direito da viewport. Os valores de **X**, **Y** e **W** devem obedecer as seguintes relações:

$$\begin{aligned}0 &\leq X, Y < 1 \\0.0001 &\leq W \leq 1 \\0 &< X + W \leq 1 \\0 &< Y + W \leq 1\end{aligned}\quad ( 2 )$$



ESCOLA POLITÉCNICA DA UNIVERSIDADE DE SÃO PAULO

Avenida Professor Mello Moraes, nº2231 CEP05508-900 São Paulo SP

---

Departamento Engenharia Mecatrônica e de Sistemas Mecânicos

Caso nenhuma viewport seja definida, o sistema assumirá  $\mathbf{x} = 0$ ,  $\mathbf{y} = 0$  e  $\mathbf{w} = 1$ .