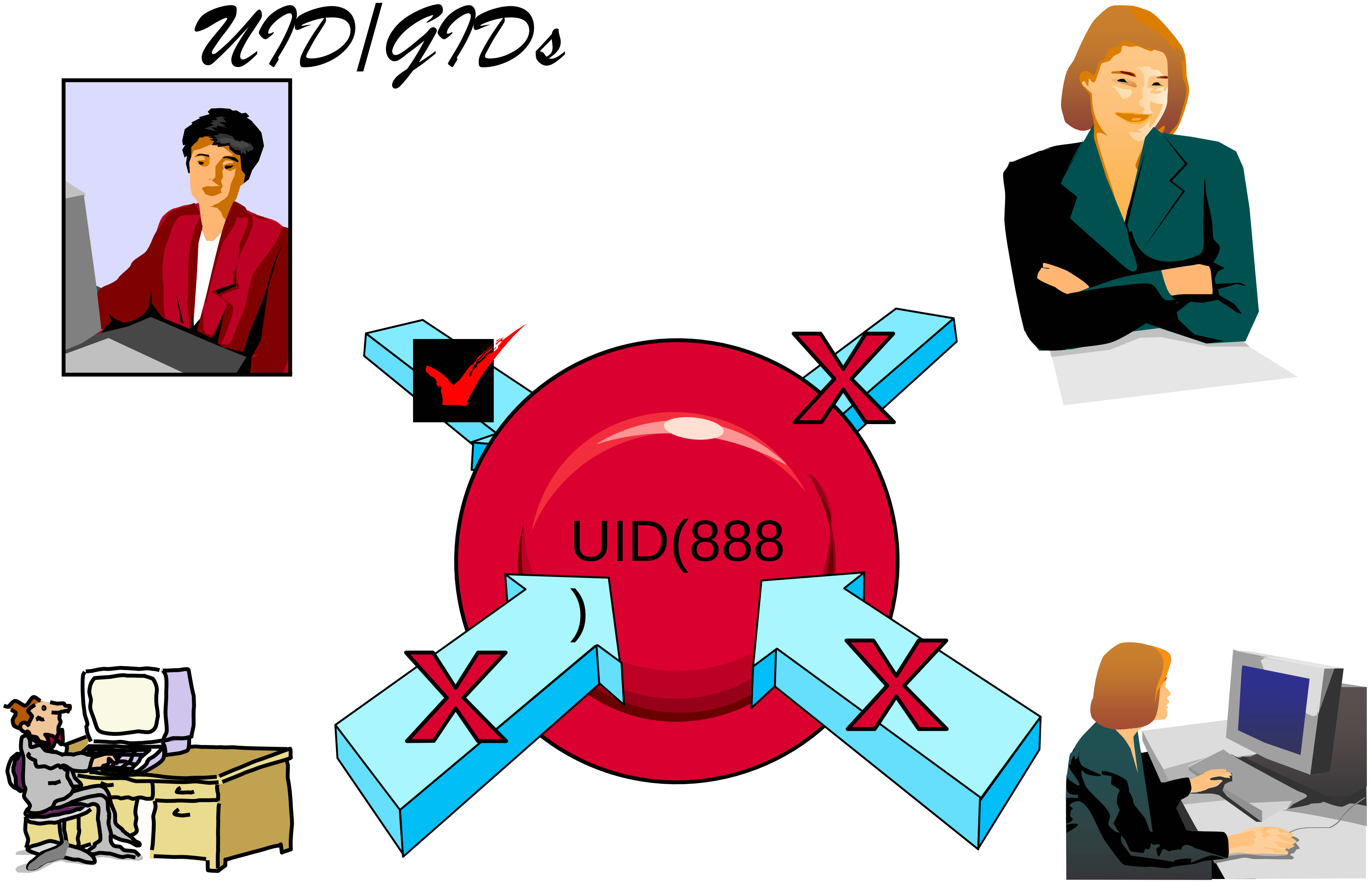
Unique UID/GID Support

Bruce R. Wells
RACF Development, IBM
845-435-7498
brwells@us.ibm.com

Agenda

- Prevention of shared UIDs and GIDs
- SEARCH enhancement for mapping UIDs and GIDs
- Automatic assignment of UIDs and GIDs
- File group ownership option

Prevention of Shared UID/GIDs



NEW!

Prevention of shared IDs

- New SHARED.IDS profile in the UNIXPRIV class
- Acts as a system-wide switch to prevent assignment of an ID which is already in use
- No generic characters allowed in name: discrete profile name must be used
- APAR OW52135
  - OS/390 2.10: UW89970 - applies to z/OS 1.1
  - z/OS 1.2: UW89971
  - z/OS 1.3: UW89972
  - In base of z/OS 1.4

http://www-1.ibm.com/servers/eserver/zseries/zos/racf/whatsnew.html

NEW!

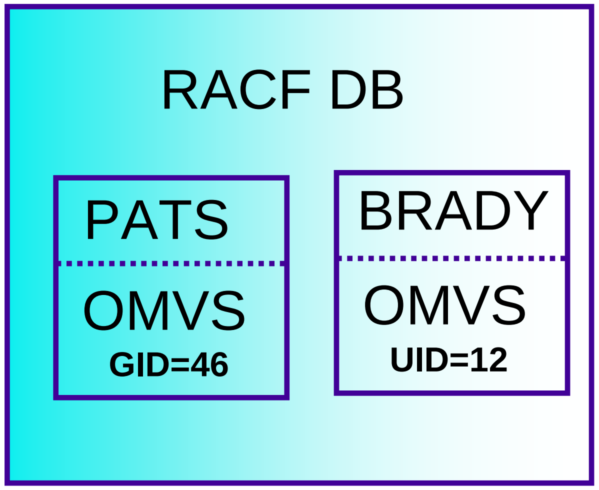
Prevention of shared IDs

- Requires AIM stage 2 or 3
- Does not affect pre-existing shared IDs
  - Customer must clean those up separately, if desired
    - Not a pre-req for using the new support
  - Can use IRRICE report to find shared UIDs
    - Can modify this report to find shared GIDs

Prevention of shared IDs ... Example

NEW!

- RDEFINE UNIXPRIV SHARED.IDS UACC(NONE)
- SETROPTS RACLIST(UNIXPRIV) REFRESH



- ADDUSER MARCY OMVS(UID(12))

IRR52174I Incorrect UID 12. This value is already in use by BRADY.

- ADDGROUP ADK OMVS(GID(46))

IRR52174I Incorrect GID 46. This value is already in use by PATS.

Prevention of shared IDs ... New SHARED keyword

NEW!

- There are valid reasons to assign a non-unique UID/GID
  - E.G. Assigning UID(0) to started task user IDS
- Do so using the new SHARED keyword in the OMVS segment of the ADDUSER, ALTUSER, ADDGROUP, and ALTGROUP commands
- SHARED requires SPECIAL, or at least READ authority to SHARED.IDS
  - Profile level audit settings can be used to log successes and failures to SHARED.IDS

Prevention of shared IDs ... Example

NEW!

RACF DB

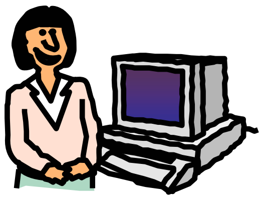
BPXOINIT
OMVS
UID=0

- PERMIT SHARED.IDS CLASS(UNIXPRIV) ID(UNIXGUY) ACCESS(READ)
- SETROPTS RACLIST(UNIXPRIV) REFRESH

UNIXGUY

```
AU OMVSKERN OMVS(UID(0) SHARED)
AG (G1 G2 G3) OMVS(GID(9) SHARED)
```

✓ OK!

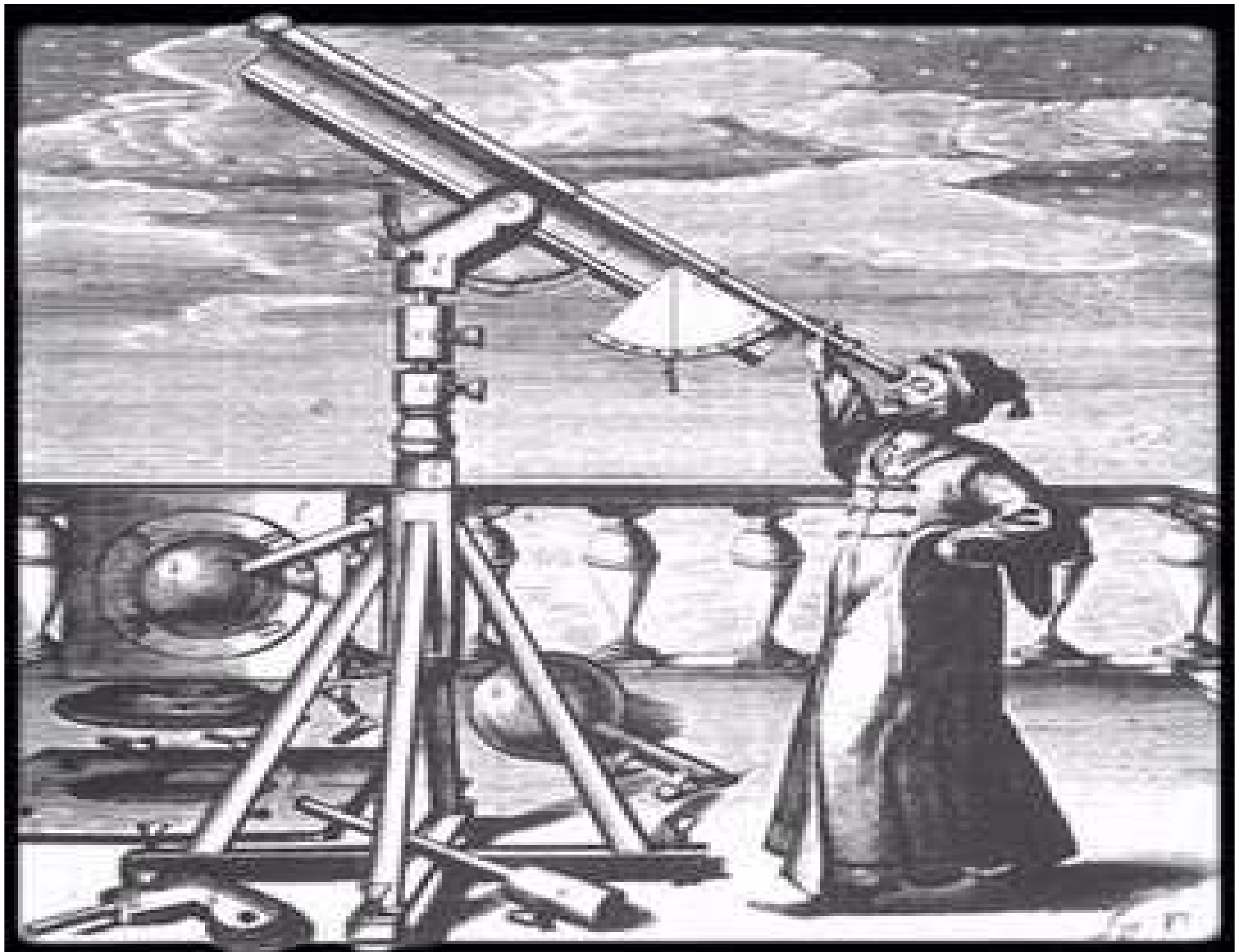


MVSGAL

```
AU MYBUDDY OMVS(UID(0) SHARED)
```

IRR52175I You are not authorized to specify the SHARED keyword.

SEARCH Enhancement



Johannes Hevelius observing with one of his telescopes (1647).

So you don't want to convert to AIM???

- Prior to OS/290 V2R10, profiles in the UNIXMAP class were used to map UIDs to user IDs and GIDs to group names
- UNIXMAP profiles automatically maintained by RACF commands
- RLIST UNIXMAP Unnn ALL shows all users with UID(nnn)
- In OS/390 V2R10, customers migrating to AIM lose that capability
- Until now!!!

Next Page

SEARCH enhancement to map UIDs and GIDs

NEW!

- SEARCH CLASS(USER) UID(0)

```
OMVSKERN
BPXOINIT
SUPERGUY
```

- SEARCH CLASS(GROUP) GID(99)

```
RACFDEV
```

- SEARCH CLASS(USER) UID(1234567)

```
ICH31005I NO ENTRIES MEET SEARCH CRITERIA
```

- Available on R4, or with OW52135

Automatic UID/GID Assignment

NEW!

Automatic UID/GID Assignment

- New AUTOUID keyword in the OMVS segment of the ADDUSER and ALTUSER commands
- New AUTOGID keyword in the OMVS segment of the ADDGROUP and ALTGROUP commands
- Derived values are guaranteed to be unique

```
ADDUSER MELVILLE OMVS(AUTOUID)

IRR52177I User MELVILLE was assigned an OMVS UID value of 4646.

ADDGROUP WHALES OMVS(AUTOGID)

IRR52177I Group WHALES was assigned an OMVS GID value of 105.
```

Automatic UID/GID Assignment ... BPX.NEXT.USER

- Uses APPLDATA of new **BPX.NEXT.USER** profile in the FACILITY class to derive candidate UID/GID values
- APPLDATA consists of 2 qualifiers separated by a forward slash ('/')
  - left qualifier specifies starting UID value, or range of UID values
  - right qualifier specifies starting GID value, or range of GID values
  - qualifiers can be null, or specified as 'NOAUTO', to prevent automatic assignment of UIDs or GIDs

Automatic UID/GID Assignment ... APPLDATA syntax

- Examples
  - RDEFINE FACILITY BPX.NEXT.USER APPLDATA('***data***')
  - good ***data***
    - 1/0
    - 1-50000/1-50000
    - NOAUTO/100000
    - /100000
    - 10000-20000/NOAUTO
    - 10000-20000/

Automatic UID/GID Assignment ... APPLDATA

- When AUTOUID or AUTOGID is issued, RACF
  1. extracts the APPLDATA from BPX.NEXT.USER
  2. parses out the starting value
  3. checks to see if it is already in use
     - If so, the value is incremented and checked again until an unused value is found
  4. assigns the value to the user or group
  5. replaces the APPLDATA with the new starting value
- The administrator can change the APPLDATA at any time using RALTER

Automatic UID/GID Assignment ... Miscellany

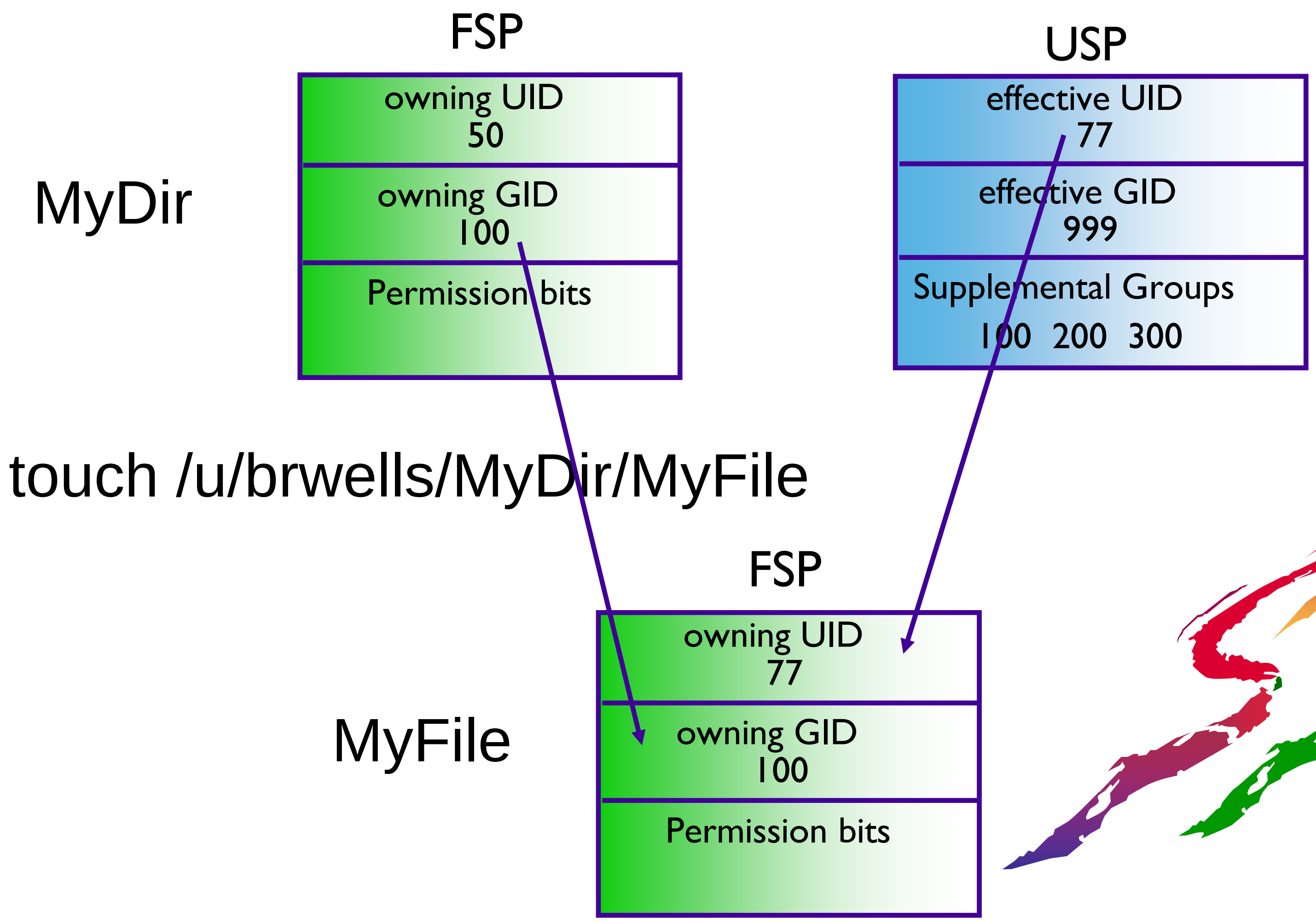
- Must be enforcing uniqueness with SHARED.IDS in order to use AUTOUID/AUTOGID
- Which in turn requires AIM stage 2 or 3

- Want an easy way to assign a unique GID to all your groups?
  - SEARCH CLASS(GROUP) NOLIST CLIST('ALTGROUP ' ' OMVS(AUTOGID)')
  - EX EXEC.RACF.CLIST

Group Ownership Option

Assigning File Group Ownership Today

- Owning UID taken from process effective UID
- Owning GID taken from parent directory



New Group Ownership Option

- Mimics 'de facto' UNIX behavior
- based on (existing, but unused) set-gid bit of parent directory
  - If set-gid on, then file group owner is set from parent directory (same as current behavior)
    - If new object is a directory, it inherits its parent's set-gid bit setting
  - If set-gid is off, then file group owner is taken from the effective GID of the process

New Group Ownership Option …

- To turn on set-gid bit for directory (nothing new here)

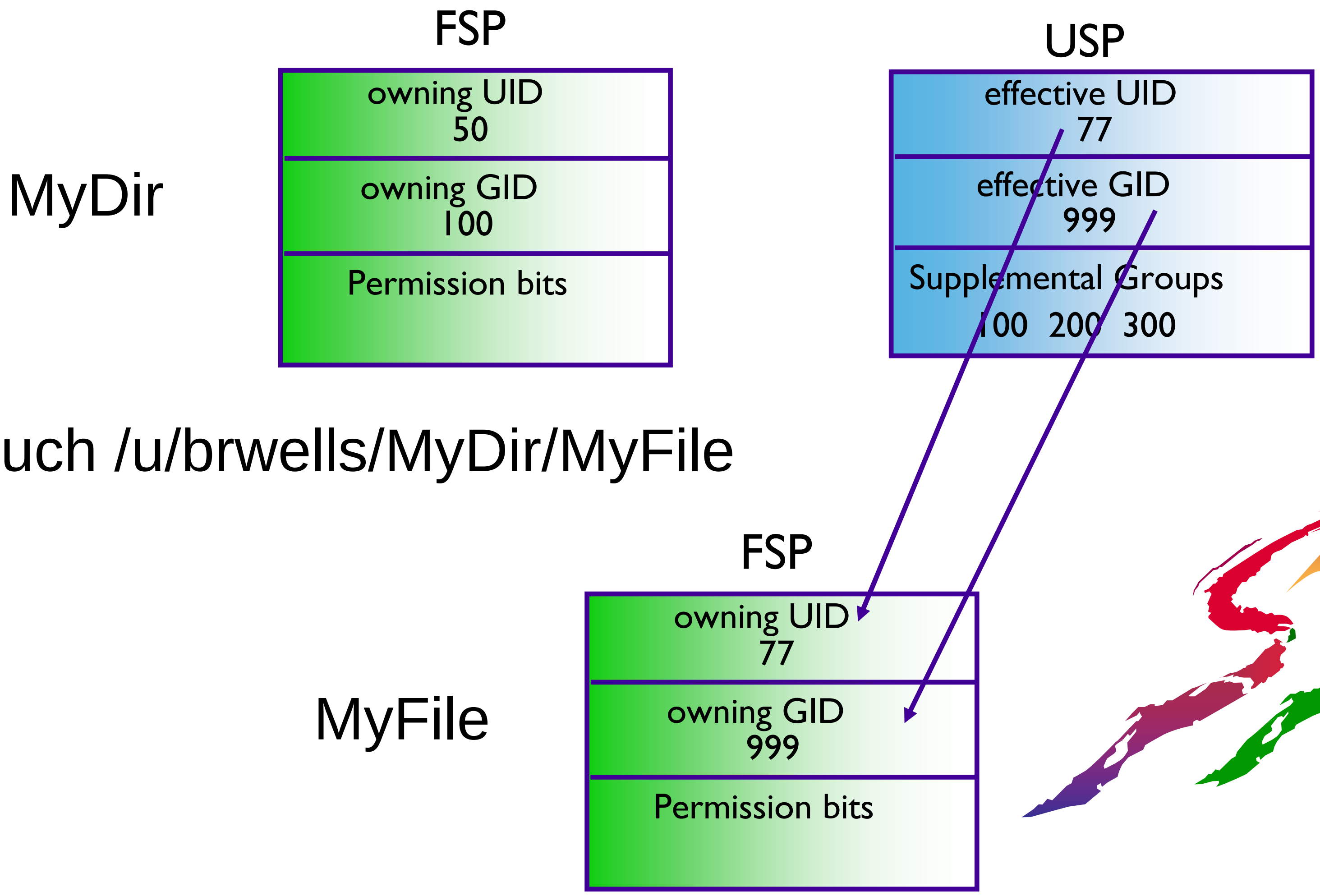
```
$ chmod g+s MyDir
$ ls -l
total 32
drwx--S---   2 BRWELLS  DEPTD60     8192 Feb  8 10:51 MyDir
drwx------   2 BRWELLS  DEPTD60     8192 Feb  8 10:51 YourDir
```

New Group Ownership Option ...

- New behavior is enabled by defining **FILE.GROUPOWNER.SETGID** in the **UNIXPRIV** class (and refreshing the UNIXPRIV class)
- Currently running processes do not recognize change
- Be aware: set-gid for new file systems will be off by default

New Group Ownership Option ...

- RDEFINE UNIXPRIV FILE.GROUPOWNER.SETGID
- SETROPTS RACLIST(UNIXPRIV) REFRESH



The End