

login DClark CompSys

<password>

who

<quit>

list

list -directory

print_wdir

change_wdir >user_dir_dir>CompSys>g

list -directory

change_wdir >system_control_dir

list

<quit>

delete accounting

listacl accounting

change_wdir >udd>CompSys>g>DClark>demo

list

qedx

r stuff

hold

setacl stuff rwa DClark

start

p

USER NAME & PROJECT
Identified for access controlPASSWORD PROTECTION
Login message; motd
Ready message desc.COMMAND LEVEL
Find cur & max load
User & project id's

Discuss QUIT

FILE SYSTEM; VIRTUAL MEMORY:
Segments, paging
Directories & sub-directories
Working dir & links
File system hierarchy
> syntax
Component names

DClark is a subdirectory.

Note access modes.

PROTECTION:
System modules part of
hierarchy.
Note access modes.No permission to delete
this file.
Explains above.

Abbreviation for user_dir_dir

ACCESS CONTROL:

Causes access violation;
no access to stuff.
Freeze state of editor;
Re-invoke command level.
Give access.

Re-tries read operation.

Print a line.

q
setacl >system_control_dir>accounting r DClark

edm blowup.pl1

```
blowup:  procedure;
         declare (i, j) fixed;
         declare a(10) fixed;
         do i = -1 to -100000 by -1;
             j = a(i);
         end;
```

w

q

pl1 blowup -table

blowup

debug

/blowup/25&t,s

i

i = 5

i

.q

list

a

call_on a

Leave editor.

Try to set access on
accounting file (fails)

PL/1 PROGRAM DEVELOPMENT:

Automatic array in stack

Will blow up running off
bottom of stack.

Makes segment

Leaves editor

Get symbol table for debugging.
Note error messages.
Invoke procedure as command.

Examine troublesome source
statement.
Print out variable.

Set variable.

Leave debugger.

DYNAMIC LINKING:

Describe call_on
Execute a as sub-routine
Discuss linking
No explicit linking needed;
linking done at first ref.
Automatic -> generality
System progs like user progs.

call_on a

Cost of little-used paths.

a

Comparison of 1st run cost
with re-run.

call_on c

c doesn't exist.

hold

edm c.pll

```
c:      procedure;
        put list ("c has been called");
        put skip;
        end;
```

.

w

q

pll c

start

Now linker finds c.

print make_list.ec

exec_com make_list abc

print abc

rename make_list.ec =.absin

enter_abs_request make_list -ag xxx

list_abs_requests

print make_list.absout

I/O STREAMS

Not paging or file system
Not reference to seg
Streams outside program,
work for any program

COMMAND MACROS: exec_com

Based on changing input stream
Note ";" in line

ABSENTEE LIKE BATCH
OR BACKGROUND:

(if finished yet)

change_wdir >udd>MPLS>Demo5

list

print fileseg

filelist fileseg

logout hold

login Demo5

<password>

print_wdir

list

print fileseg

howmany a

howmany b

logout

SUBSYSTEM PROTECTION:
Subsystem is group of programs
encapsulated together in
ring of protection.

Explain collection of "grades"

Can access from ordinary
process.
Note ACL's.

Under-privileged user,
comes up in Ring 5.

Note access.

Can't do it from here.

Access selected info
through special software.

That's all, folks!