K. T. Pogran
8/30/72

Multics "Prepared Demo"

The Multics prepared demo was developed over a period of time by David Clark, a graduate student with the Computer System Research Division of Project MAC. It takes about one hour to run through this demo.

Concepts covered include:

1) Command language
2) File system
3) Virtual memory
4) Access control
5) Error recovery
6) Creation, compilation, execution, and debugging of programs
7) Dynamic linking to called subroutines
8) Input/Output flexibility
9) Command macros and batch (or background) processing
10) Sub-system privacy and protection

The enclosed "demonstrator's script" gives the lines typed into Multics, along with notes giving the features exemplified, and reminders for further elucidation by the demonstrator. Following the demonstrator's script is a sample console session, with lines typed by the user underlined.

login Pogran CompNet	USER NAME & PROJECT
<password>	PASSWORD PROTECTION Login message;motd Ready message desc.
who	Find cur & max load User & project id's
<quit>	Discuss QUIT
list	FILE SYSTEM; VIRTUAL MEMORY: Segments, paging Directories & sub-directories Working dir & links File system hierarchy ">" syntax Component names
list -directory	
print_wdir	
change_wdir >user_dir_dir>CompNet	Pogran is a subdirectory.
list -directory	Note access modes.
change_wdir >system_control_dir	PROTECTION: System modules part of hierarchy. Note access modes.
list	
<quit>	
delete accounting	No permission to delete this file.
listacl accounting	Explains above.
change_wdir >udd>CompNet>Pogran>demo	Abbreviation for user_dir_dir
list	ACCESS CONTROL:
qedx	A text editor.
r stuff	Causes access violation; no access to stuff.
hold	Freeze state of editor; Re-invoke command level.
setacl stuff rwa Pogran	Give access.
start	Re-tries read operation.
p	Print a line.
q	Leave editor.

```
setacl >system_control_dir>accounting r Pogran
```
Try to set access on accounting file (fails)

```
create xyz
```
I/O SYSTEM:
Not paging or file system. Referencing segs is not I/o. Neither is paging from disk, drum. I/O over streams, re-directed outside program, so works for any command. Note ";" in command line. Look at results

```
print_attach_table
```

```
file_output xyz;print_attach_table;console_output
```

```
print xyz
```

```
dprint -delete -copy 2 xyz
```
Queued offline printing for bulky output

```
edm blowup.pl1
```
PL/1 PROGRAM DEVELOPMENT: (Using another text editor)

```
blowup:    procedure;

           declare (i, j) fixed;

           declare a(10) fixed;
```
Automatic array in stack

```
           do i = -1 to -100000 by -1;

               j = a(i);
```
Will blow up running off bottom of stack.

```
           end;
```

```
.
```

```
w
```
Makes segment

```
q
```
Leaves editor

```
pl1 blowup -table
```
Get symbol table for debugging

```
blowup
```
Note error messages. Invoke procedure as command.

```
debug
```

```
/blowup/25&t,s
```
Examine troublesome source statement.

```
i
```
Print out variable.

```
i = 5
```
Set variable.

```
i
```

```
.q
```
Leave debugger.

list

a

print call_on.pl1

call_on a

a

call_on c

hold

edm c.pl1

```
c:         procedure;

           put list ("c has been called");

           put skip;

           end;
```

.

w

q

pl1 c

start

print print_who.ec

exec_com print_who abc

print abc

rename print_who.ec =.absin

enter_abs_request print_who -ag xxx

list_abs_requests

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.
Cost of little-used paths.
Comparison of 1st run cost
with re-run.

c doesn't exist.

Now linker finds c.

COMMAND MACROS: exec_com
Based on changing input stream

ABSENTEE LIKE BATCH
OR BACKGROUND:

```
print print_who.absout
```

(if finished yet)

```
change_wdir >udd>MPLS>Demo5
```

SUBSYSTEM PROTECTION:
A subsystem can be a
group of programs
enscapulated together in
a ring of protection.

```
list
```

```
print fileseg
```

Explain collection of "grades

```
filelist fileseg
```

Can access from ordinary
process.
Note ACL's.

```
logout hold
```

```
login Demo5
```

```
<password>
```

Under-privileged user,
comes up in Ring 5.

```
print_wdir
```

```
list
```

Note access.

```
print fileseg
```

Can't do it from here.

```
howmany a
```

Access selected info
through special software.

```
howmany b
```

```
logout
```

That's all, folks!

```
Multics 17.6b; MIT, Cambridge, Mass.
Load = 31.5 out of 50.0 units; users = 32
login Pogran CompNet
Password:
You are protected from preemption.
Pogran CompNet logged in: 08/30/72  1414.8 edt Wed from 1050 terminal "."
Last login 08/30/72  1355.9 edt Wed from 1050 terminal "."
new/updated help segs: area.changes, v2pl1.lang, login, bcpl_library, gls.
r 1415  2.757  0+62

who

Multics 17.6b, load 32.5/50.0; 33 users
Absentee users 1/2

Roach.SysMaint*
IO.SysDaemon
Backup.SysDaemon
Network_Daemon.CNet
Carignan.IPCUS
Downing.Multics
Garman.Mul
QUIT
r 1415  .629  5+10

list

Segments= 12, Records= 18.

r wa    1   demo2
r wa    1   demo1.runoff
re      2   net_who
rewa    2   net_who.pl1
r wa    2   imp_peek.pl1
re      2   imp_peek
r wa    1   file_listing.absin
rewa    1   dump8.pl1
rewa    1   dump8
re      2   cnet_report
            print_report
rewa    2   atc_sim.pl1
rewa    1   atc_sim

r 1415  3.327  9+83

list -directory

Directories= 3, Records= 4.

rewa    1   demo
rewa    1   graphics
            g
rewa    2   cnet
            c

r 1416  .902  5+24
```

```
print_wdir
>user_dir_dir>CompNet>Pogran
r 1416   .370   6+18


change_wdir >user_dir_dir>CompNet
r 1417   .392   5+15


list -directory

Directories= 16, Records= 40.

rewa    5    Ryan
             ndr
re      2    Tavares
             cdt
re      1    Bressler
re      1    RMetcalfe
rewa    3    simulate
             sim
re      2    imp_dim
             r0
             imp_dim_
rewa    2    log
rewa    4    library
             lib
             include
rewa    1    demo
rewa    3    Pogran
             ktp
             Network_Consultant
rewa    4    Padlipsky
             map
rewa    2    Network_Daemon
             nd
re      2    Meyer
             ewm
re      5    Kanodia
             rkk
re      2    DWells
             dmw
re      1    Bhushan
             akb


r 1417   2.826   7+32


change_wdir >system_control_dir
r 1418   .320   3+12


list

Segments= 34, Records= 368.

        15    sat
        11    10__accounting
r wa     1    net_log
         2    lines.backup
r        2    lines
        42    pnt
        34    accounting
        52    delete.10__accounting.-.08/29/72
r       17
```

`r 1418  .331  5+11`

`delete accounting`

```
delete: accounting is protected.  Do you want to delete it?  yes
delete: Access is incorrect on directory containing accounting.
r 1420  1.413  39+49
```

`listacl accounting`
```
rwa       .sys_control.*
rwa       *.SysAdmin.*
```

`r 1428  1.156  2+35`

`change_wdir >udd>CompNet>Pogran>demo`
`r 1428  .316  0+16`

`list`

`Segments= 6, Records= 6.`

```
         :. stuff
r wa     :. print_who.ec
re       :. call_on
r wa     :. call_on.pl1
re       l. a
r wa     l. a.pl1
```

`r 1428  .814  5+24`

`qedx`
`r stuff`

```
Error: Improper access by qedx$qedx|1304
(>system_library_standard>bound_qedx_)
(while in pl1 operator cs_move)
referencing >udd>CompNet>Pogran>demo>stuff|0
(offset is relative to base of segment)
r 1429  3.516  15+112
```

`hold`
`r 1430  .358  7+29`

`setacl stuff rwa Pogran`
`r 1430  .578  6+26`

`start`
`p`
`This is some dsample text.`
`q`
`r 1430  .712  26+26`

`setacl >system_control_dir>accounting r Pogran`
`setacl: Access is incorrect to the segment or containing directory. >system_contro`
`r 1431  .659  5+32`

```
create xyz
r 1432  .643  5+29

print_attach_table

Stream - Type - Device

user_output syn user_i/o
user_input syn user_i/o
user_i/o tw_ tty170
error_output syn user_i/o
r 1432  .584  5+18

file_output xyz; print_attach_table; console_output
r 1433  1.386  5+30

print xyz


                    xyz        08/30/72  1433.7 edt Wed


Stream - Type - Device

user_output syn file_output_stream_
user_input syn user_i/o
user_i/o tw_ tty170
error_output syn user_i/o
file_output_stream_ file_ >udd>CompNet>Pogran>demo>xyz


r 1433  1.262  5+31

dprint -delete -copy 2 xyz
1 request signalled, 3 already queued.
r 1434  1.789  5+40

edm blowup.pl1
Segment not found.
Input.
blowup:    procedure;
           declare (i, j) fixed;
           declare a(10) fixed;
           do i = -1 to -100000 by -1;
                j = a(i);
           end;
.
Edit.
w
q
r 1435  2.485  86+50
```

```
.pll blowup -table
PL/1


ERROR 101, SEVERITY 1 IN STATEMENT  1 ENDING ON LINE     7
The source program had insufficient end statements. One has been
supplied.


ERROR 307, SEVERITY 1
Warning: The variable "a" has been referenced but has never been set.

r 1436  7.045  16+290

bloe#wup

Error: out_bounds_err by >udd>CompNet>Pogran>demo>blowup$blowup|25
referencing stack_4|777777 (in process dir)
r 1436  1.380  2+47

debug
/blowup/25&t,s
                 j = a(i);
i
   1410      110            -778
i = 5
Changing 204|1410
777777776366 to 000000000005
i
   1410      110                  5
.q
r 1437  2.882  94+97

list

Segments= 8, Records= 8.

re       1   blowup
r wa     1   blowup.pll
r wa     1   stuff
r wa     1   print_who.ec
re       1   call_on
r wa     1   call_on.pll
re       1   a
r wa     1   a.pll

r 1437   .883  5+28


a
a has been called
r 1438  2.299  10+60
```

```
print call_on.pll
.
                    call_on.pll          08/30/72   1438.5 edt Wed


call_on:   procedure (which);

           declare which char (*);

           declare (a, b, c) entry;

           if which = "a" then call a();
           else if which = "b" then call b();
           else if which = "c" then call c();

           end call_on;


r  1438   .523   5+12

call_on a
a has been called
r  1439   .426   5+21

a
a has been called
r  1439   .171   5+0

call on c

Error: Linkage error by >udd>CompNet>Pogran>demo>call_on$call_on|75
Referencing c|c.
Segment not found.
r  1439   1.053   5+49

hold
r  1439   .146   6+2

edm c.pll
Segment not found.
Input.
c:          procedure;
            put list ("c has been called");
            put skip;
            end;
.
Edit.
w
q
r  1440   1.650   59+41

pll c
PL/I
r  1440   2.969   6+161

start
c has been called
r  1440   .593   3+28
```

```
print print_who.ec

                     print_who.ec          08/30/72  1440.9 edt Wed


  file_output &1
  who
  console_output


  r 1440  .395  2+10

exec_com print_who abc
file_output abc
r 1441  2.051  4+34

print abc


                  abc          08/30/72  1441.9 edt Wed


who

Multics 17.6b, load  36.0/50.0; 36 users
Absentee users 1/2

Roach.SysMaint*
IO.SysDaemon
Backup.SysDaemon
Network_Daemon.CNet
Carignan.IPCUS
Downing.Multics
Garman.Multics
Hill.Janus

QUIT
r 1442  .888  10+18

rename print_who.ec =.as#bsin
r 1442  .567  6+29

enter_abs_request print_who -ag xxx
23 already requested.
r 1442  5.258  8+131

list_abs_requests

Queue 3:    0 requests. 23 total requests.

r 1443  2.572  4+65

print print_who.absout


                  print_who.absout    08/30/72  1443.3 edt Wed



Absentee user Pogran CompNet logged in: 08/30/72  1442.9 edt Wed
new/updated help segs: area.changes, v2pl1.lang, login, bcpl_library, gls.
r 1443  2.819  0+52
```

```
QUIT
r 1446  .260  4+11

logout


Pogran CompNet logged out 08/30/72  1447.0 edt Wed
CPU usage 1 min 16 sec
hangup
```

N.B. — Protected sub-system demo
not included in this draft.

KJP
8/30/72