

Michael Kepple  
Operating Systems

Lab Ten

1) Add hook **for** msg\_dmp() in /usr/src/servers/is/dmp.c

Diff output:

26d25

```
<      { F9,      msg_dmp, â\200\234Message Matrix printoutâ\200\235 },
```

2) Add msg\_dmp() function declaration in proto.h

Diff output:

13d12

```
<  void msg_dmp(void);
```

3) GET\_MSGS defined in /usr/src/include/minix/

Diff output:

475d474

```
< #      define GET_MSGS          25      /* get system message matrix */
```

4) MATRIX\_SIZE, MAX\_INDEX defined in /usr/src/kernel/const.h

Diff output:

11,14d10

```
<
```

```
< #define MATRIX_SIZE 10
```

```
< #define MAX_INDEX 9
```

```
<
```

5) msgMatrix defined in /usr/src/kernel/glo.h

Diff output:

27d26

```
<  EXTERN int msgMatrix[10][10];
```

6) msgMatrix declared (memset to 0) in /usr/src/kernel/proc.c (proc\_init())

Diff output:

119d118

```
<      memset(&msgMatrix, 0, sizeof(msgMatrix));
```

7) edited /usr/src/kernel/system/do\_getinfo.c to **return** msgMatrix as message to caller.

Diff output:

53,57d52

```
<      case GET_MSGS: {
<          length = 100*sizeof(int);
<          src_vir = (vir_bytes) &msgMatrix;
<          break;
<      }
```

8) msg\_dmp() function defined in /usr/src/servers/is/dmp\_kernel.c

Diff output:

60,82d59

```
<  void msg_dmp()
<  {
<      int I, h, tempMatrix[MATRIX_SIZE][MATRIX_SIZE];
<      sys_getmsginfo(&tempMatrix);
<      if ((I = sys_getproctab(proc)) != OK)
<      {
<          printf(â\200\234IS: Warning: couldn't get copy of process table %d\nâ\200\235
<      , i);
<      }
<      printf(â\200\234\n Sent From                                Received @:\nâ\200\235);
<      printf(â\200\234%14s%05d %05d %05d %05d %05d %05d %05d %05d %05d %05d\nâ\200\235
<      , &â\200\234â\200\235,0,1,2,3,4,5,6,7,8,-2);
```

```

<  for (i=0; i<MAX_INDEX; i++)
<  {
<      printf(â\200\234\n%7s(%d) â\206\222 â\200\234, proc_name(i),i);
<      for (j=0; j<MATRIX_SIZE; j++)
<          printf(â\200\234%05d â\200\234, tempMatrix[i][j]);
<      }
<      printf(â\200\234\n%6s(-2) â\206\222 â\200\234, proc_name(SYSTEM));
<      for (i=0; i<MATRIX_SIZE; i++)
<          printf(â\200\234%05d â\200\234, tempMatrix[MAX_INDENT][i]);
<  }
<

```

9) system task calls to other endpoints tracked in /usr/src/kernel/system.c kernel\_call\_finish().

Diff output:

70,73d69

```

<      int callBack = caller->p_endpoint;
<      if (callBack >= 0 && callBack <= MAX_INDEX)
<          msgMatrix[MAX_INDEX][caller->p_endpoint]++;
<

```

10) system task call from other endpoints tracked in /usr/src/kernel/system.c kernel\_call()

Diff output:

148,150d143

```

<      int callee = caller->p_endpoint;
<      if (callee >= 0 && callee < MATRIX_SIZE)
<          msgMatrix[callee][MAX_INDEX]++;
<

```

11) all other endpoint messages tracked in /usr/src/kernel/proc.h

Diff output:

504,510d503

```

<      if (result == OK)
<      {
<          int caller = proc_nr(caller_ptr);
<          int callee = src_dst_e;
<          if ((caller >= 0 && callee >=0) && (callee < MAX_INDEX && caller < MAX_I
NDEX))
<              msgMatrix[caller][callee]++;
<      }
<

```

12) make, make install in all local directories, make install in /usr/src/releasetools.

13) 'shutdown -r' to reboot the machine.

\* according to /usr/src/include/minix/com.h, endpoint 7 = Minix root filesystem