HSSPOOL

Ross Systems International

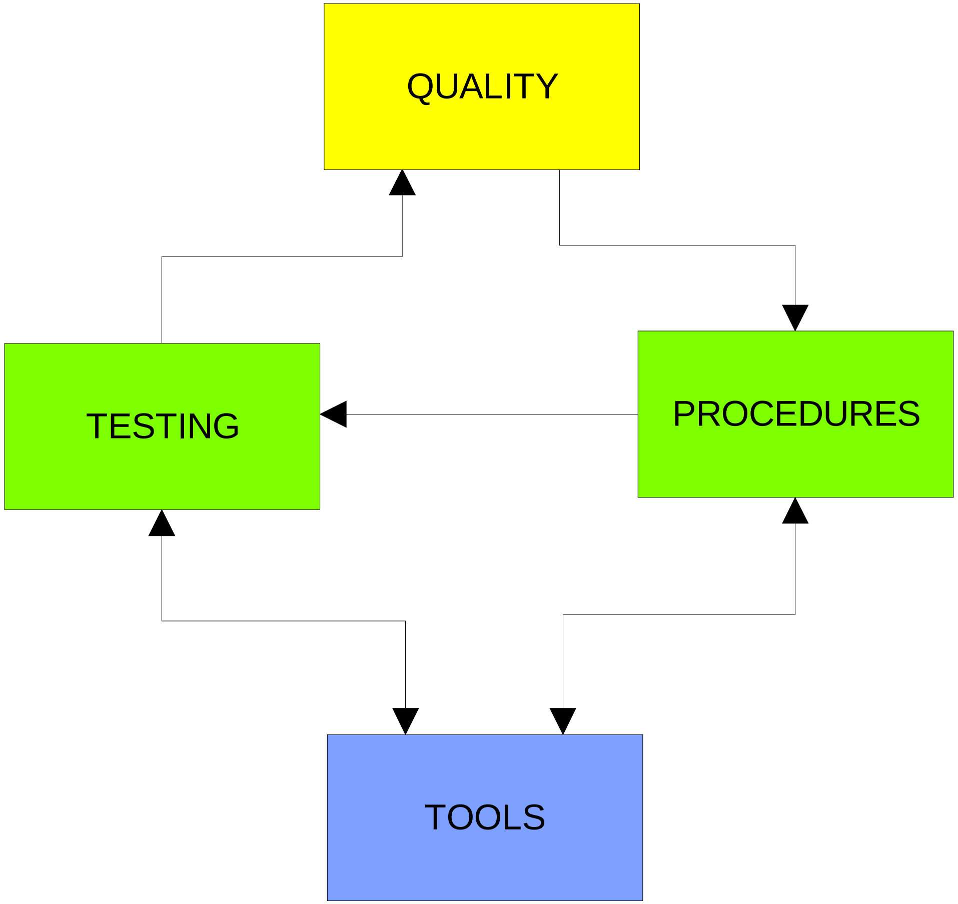
Rupert Stanley

WHO WE ARE

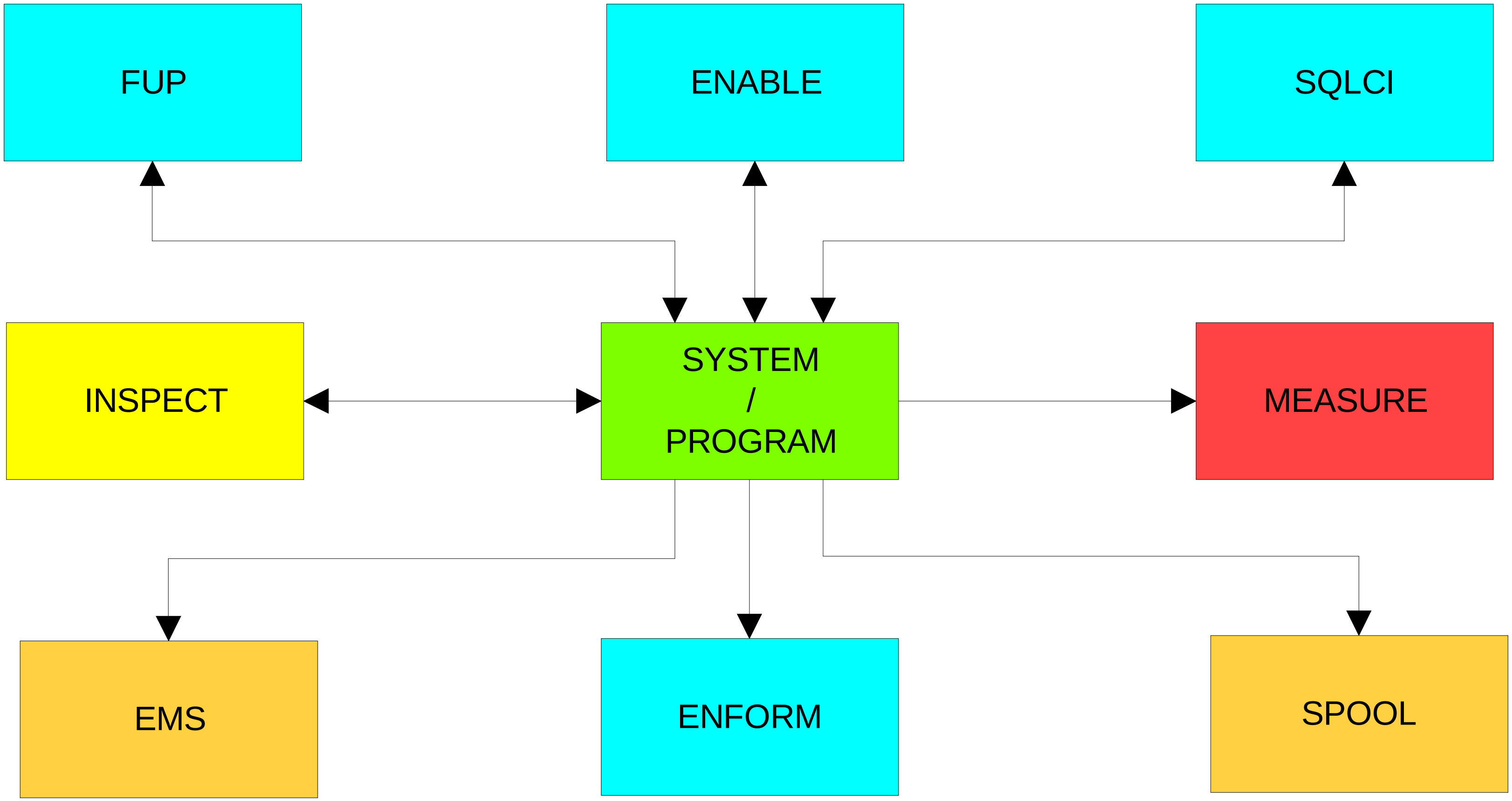
SMALL SOFTWARE HOUSE SPECIALISING IN NONSTOP:

1. TESTING TOOLS
2. BANKING SECURITY
3. COMMUNICATIONS
4. SYSTEMS PROGRAMMING

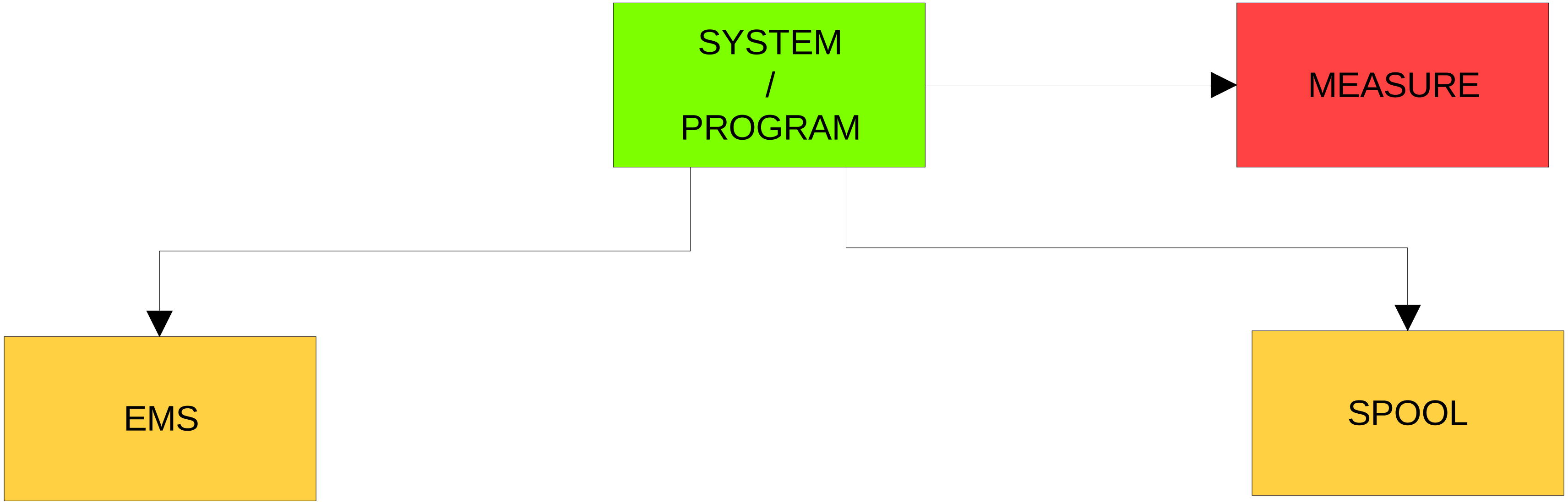
NEED FOR HSSPOOL



AVAILABLE TOOLS

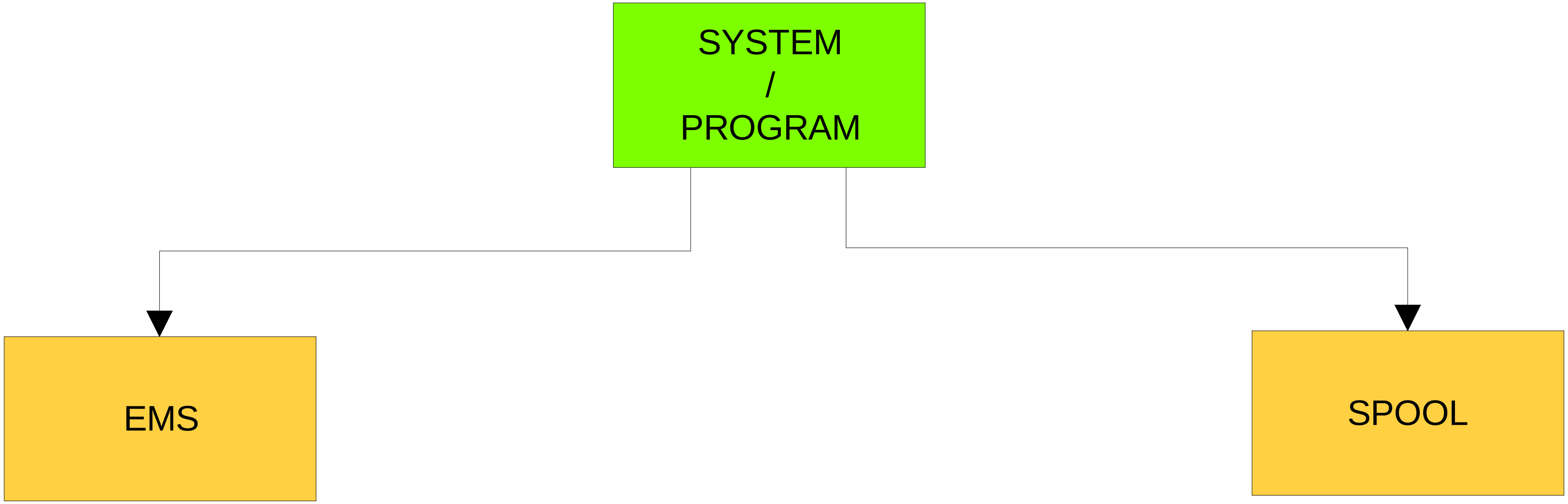


HOW DOES YOUR SYSTEM PERFORM IN REAL TIME



IN DETAIL, BY TRANSACTION CONTENTS?

HOW DOES YOUR SYSTEM PERFORM IN REAL TIME



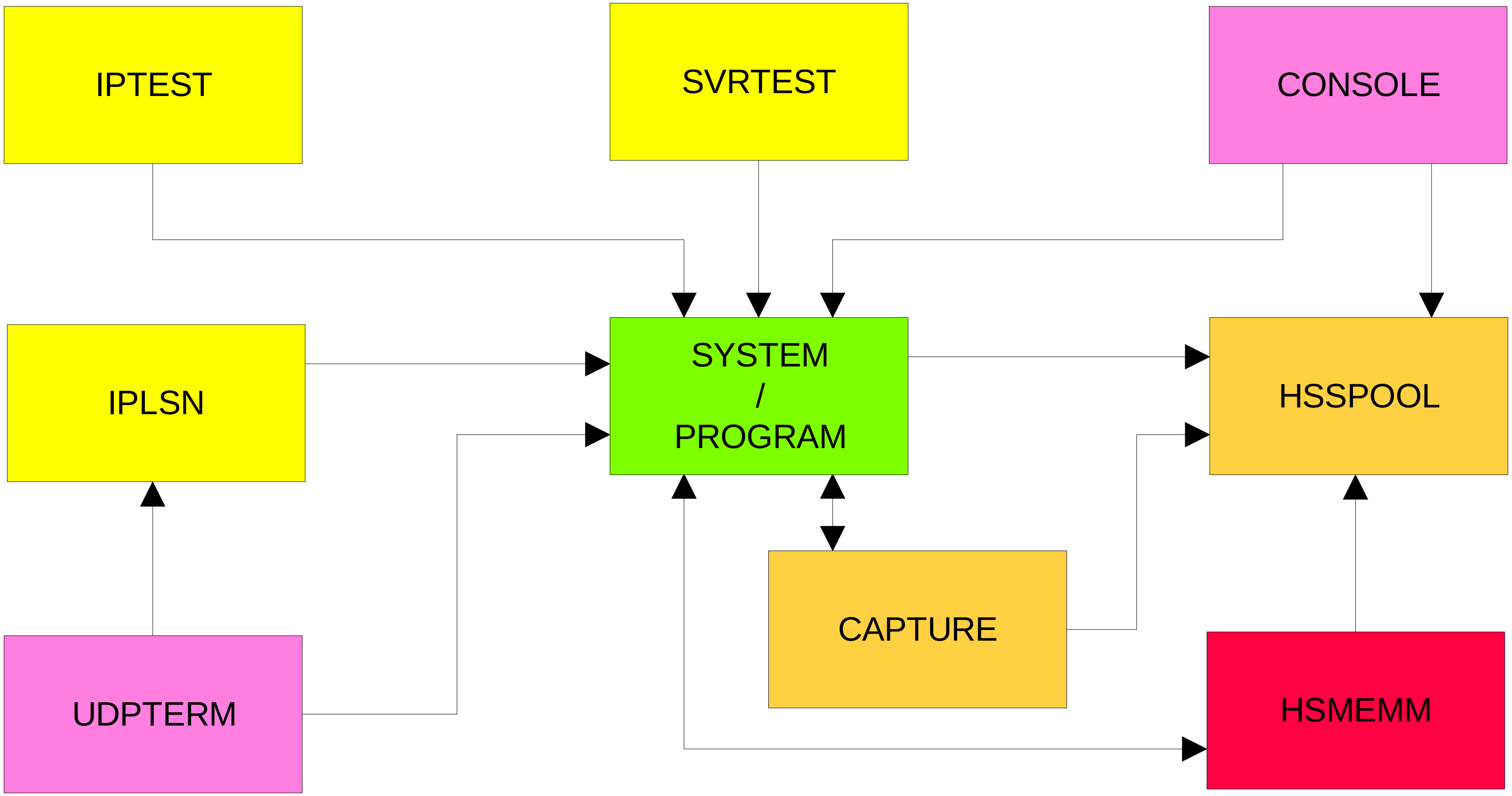
I MEANT REAL TIME WITHOUT IMPACTING THE SYSTEM

HOW DOES YOUR SYSTEM PERFORM IN REAL TIME

SYSTEM
/
PROGRAM

THERE DON’T SEEM TO BE ANY TOOLS LEFT

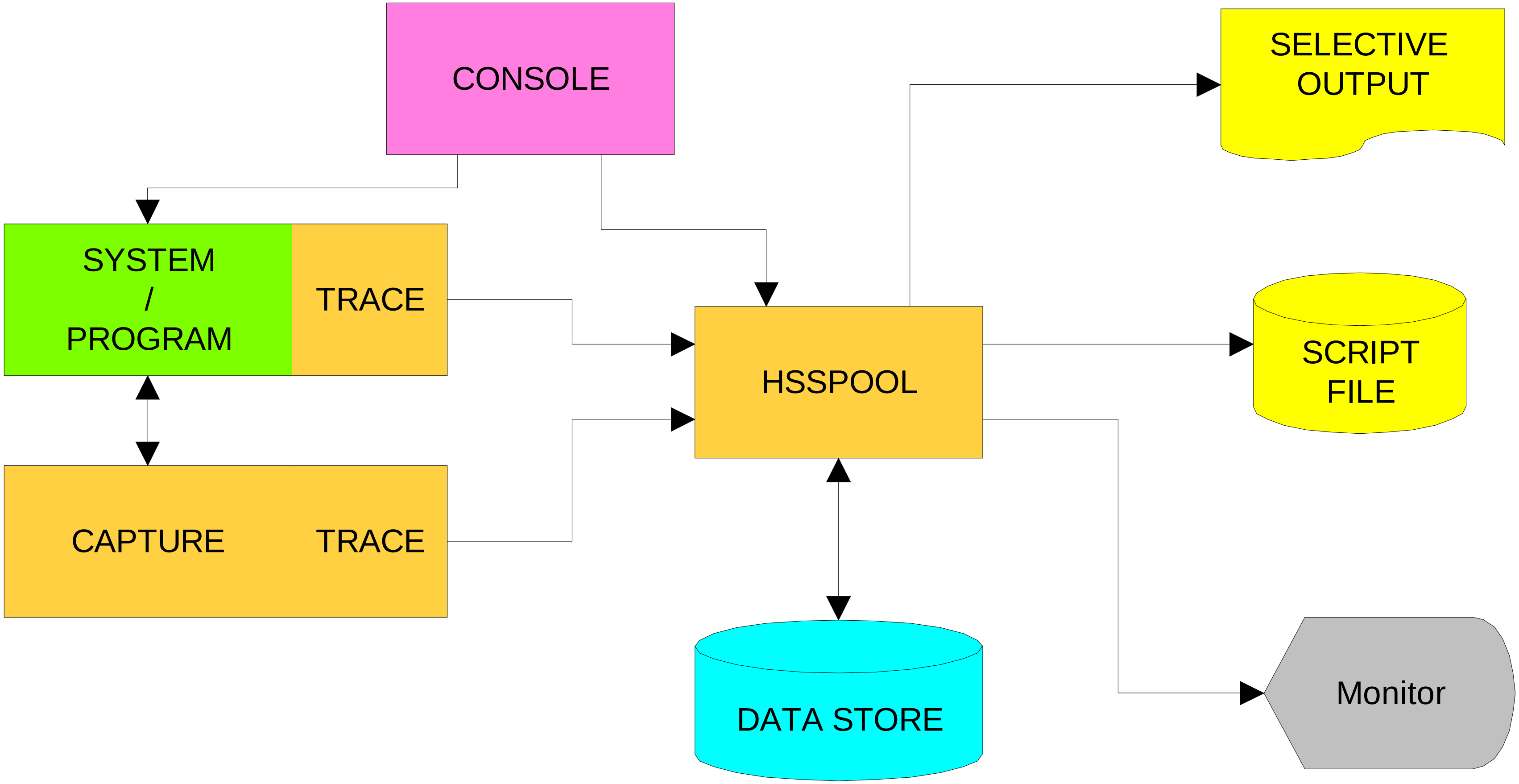
WHAT DO WE NEED?



WHAT DO THEY DO?

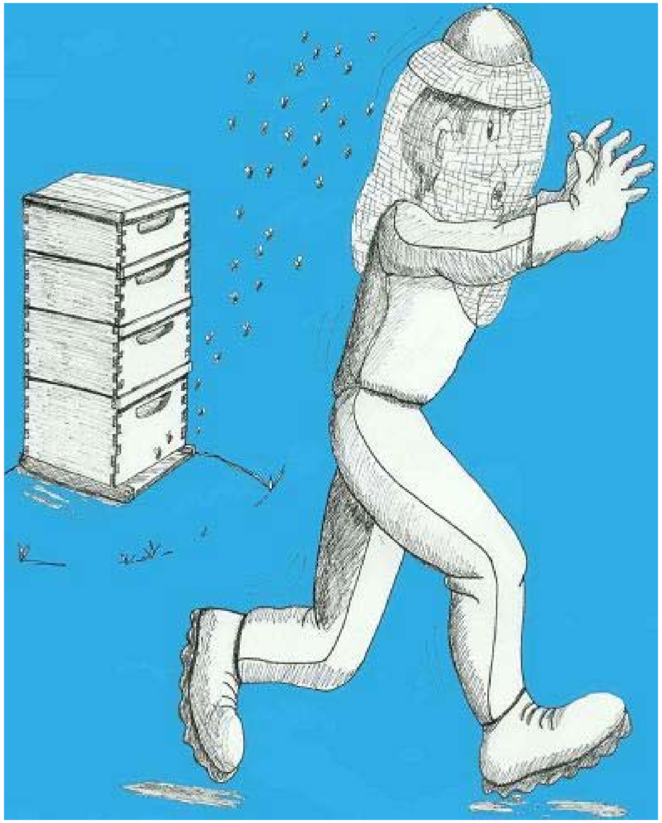
1.	SVRTEST	Scripted Pathway/$RECEIVE Interface
2.	IPTEST	Scripted TCP/IP Interface
3.	IPLSN	General Purpose TCP/UDP IP Interface
4.	CAPTURE	Transaction Capture Program
5.	HSEMM	HSM Emulator, Performs HSM Cryptography and Provides Complete Cryptographic Traces
6.	HSSPOOL	High Speed Data Capture Process
7.	CONSOLE	Control Interface for Tools

HSSPOOL, WHAT IT DOES



WHAT’S THE PROBLEM?

IF WE LOOK INTO THE HIVE TO SEE HOW MUCH HONEY THERE IS THIS IS WHAT HAPPPENS - ANGRY BEES! NO HONEY!



I.E.

MEASUREMENTS

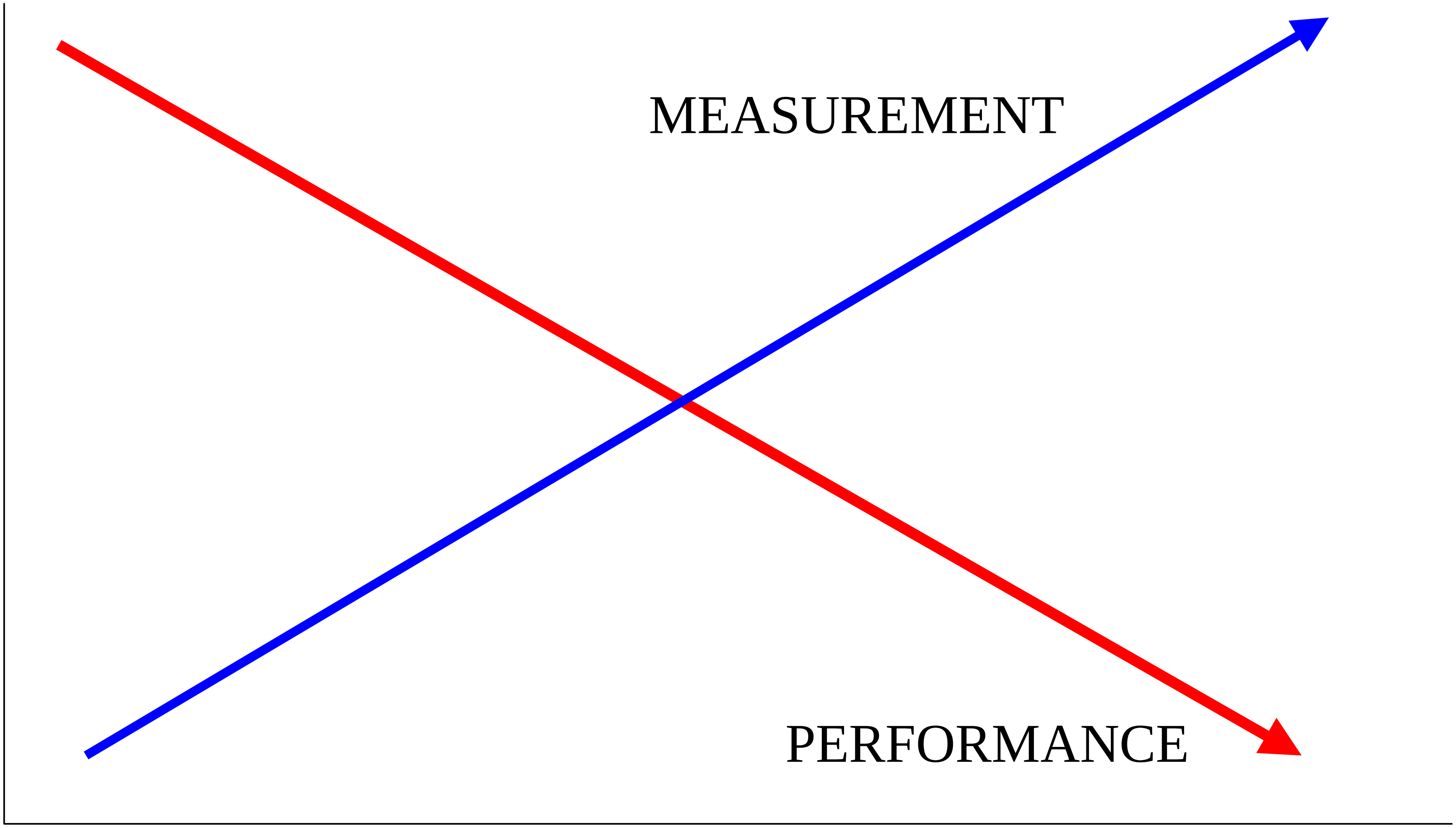
INTERFERE

WITH

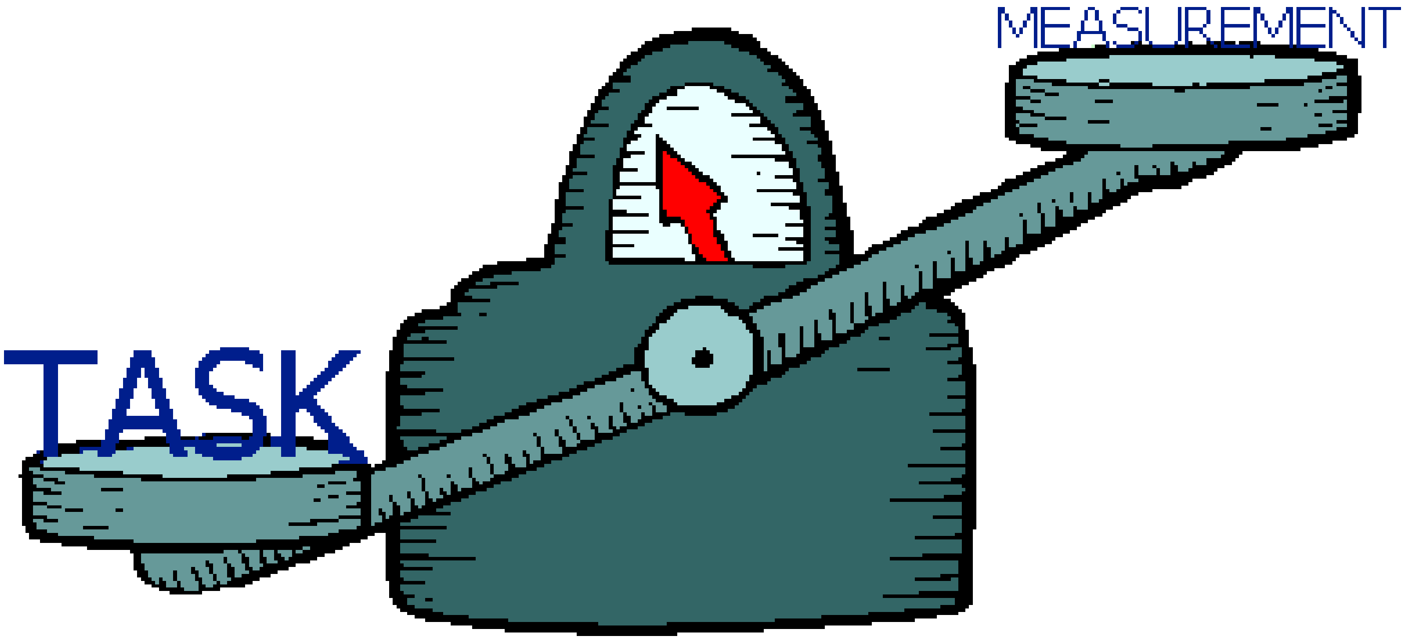
EXPERIMENTS

SO

PERFORMANCE PROBLEM

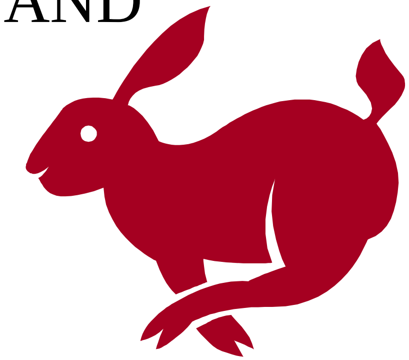


WHAT WE NEED



HSSPOOL NEEDS TO BE

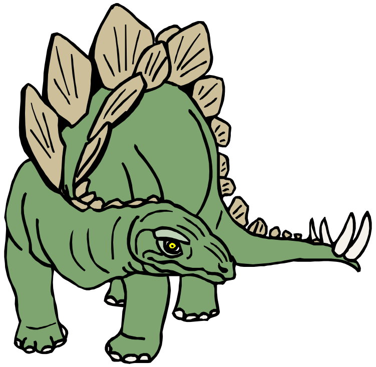
FAST AND



LIGHT

NOT

HEAVY AND



SLOW

AT LEAST 1000 TPS

AND SIMPLE TO USE, WAITED WRITES PLEASE
(i.e. Ordinary WRITE with Wait for I/O completion)

HSSPOOL TPS

S SERIES DATA COLLECTION TPS MUST BE GREATER THAN

1000 TPS

TESTS SHOW PERFORMANCE OF

K SERIES	UP TO 900 TPS
S SERIES	UP TO 2400 TPS

AND THIS CAN BE MAINTAINED

DON’T ALSO FORGET

WE NEED TO
MONITOR OUT
SYSTEMS WITH

MANY DIFFERENT COMPONENTS WORKING TOGETHER

INTERRELATIONS ARE IMPORTANT

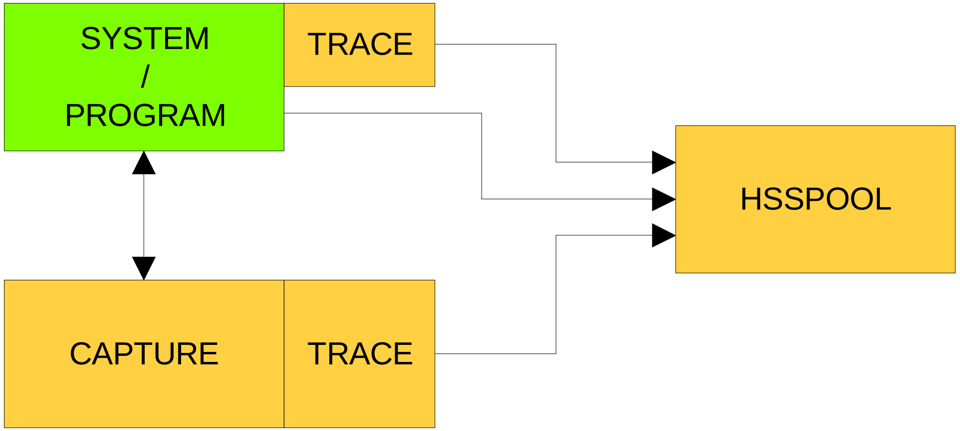
WE NEED TO KNOW:

1. WHERE PROCESSES ARE GENERATING DATA
2. WHAT IS HAPPENING IN THE PROCESSES
3. WHEN IT HAPPENS

IN ORDER TO FIND OUT

4. WHY THE SYSTEM IS OPERATING AS IT DOES

HSSPOOL INPUT



FROM MANY DIFFERENT SOURCES CONCURRENTLY

1. TEXT
2. DUMPS
3. TOKENISED EMS

INPUT COMPONENTS

CAPTURE –

PROCESS TO SIT ON THE INTER-PROCESS MESSAGE INTERFACE BETWEEN TWO PROCESSES AND CAPTURE THE MESSAGE TRAFFIC DATA:

1. COMMAND SENT
2. RESPONSE RECEIVED
3. TRANSACTION TIME

TRACE -

C API TO ENABLE THE SIMPLE INSERTION OF TRACE CODE IN A PROGRAM. IT HAS FUNCTIONS FOR SENDING:

1. TEXT MESSAGES
2. ANNOTATED MEMORY DUMP MESSAGES
3. CAPTURE MESSAGES

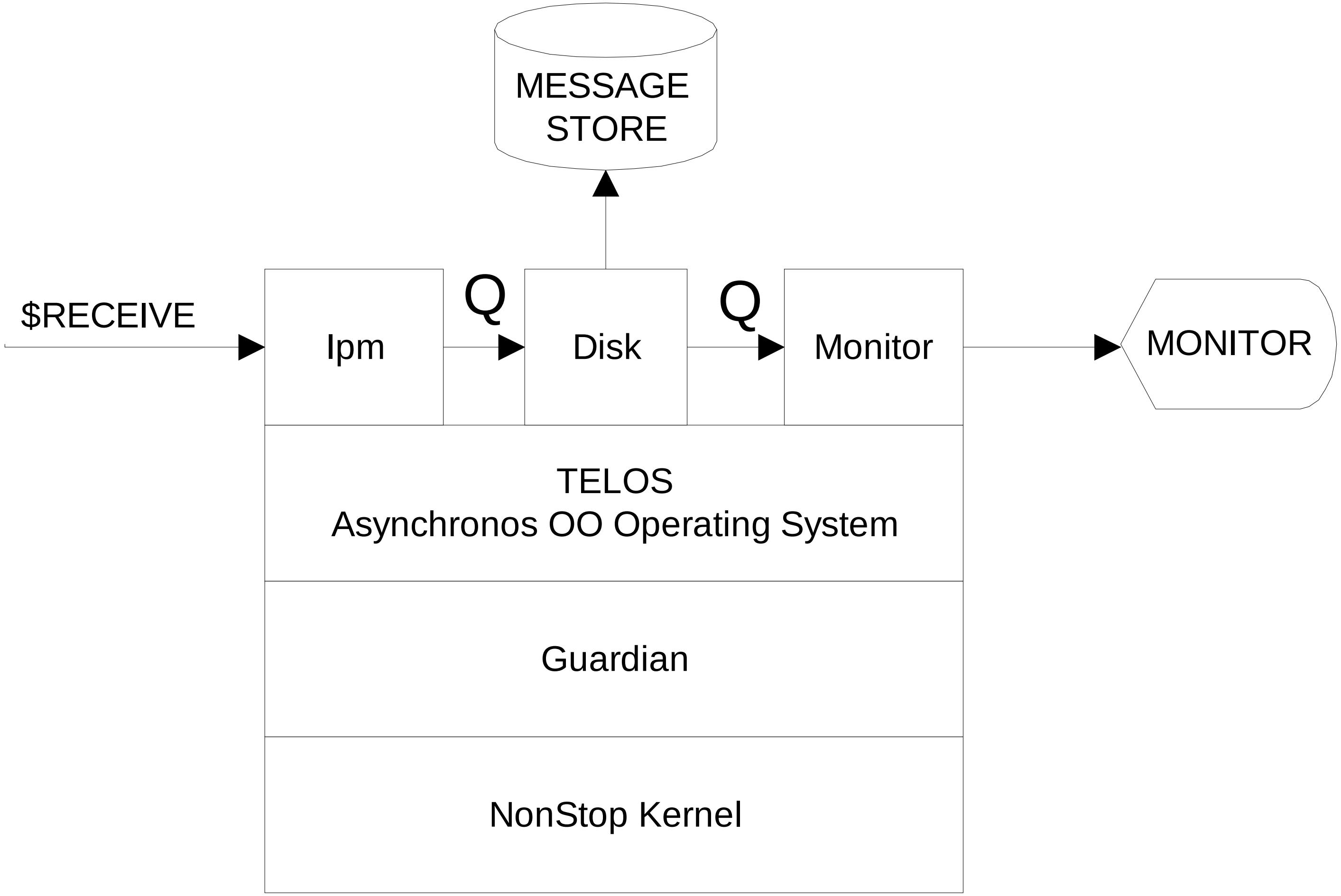
HSSPOOL MONITOR OUTPUT

HSSPOOL
Monitor

1. TERMINAL AS TEXT

2. EMS COLLECTOR

PERFORMANCE BUILT IN



IPM MESSAGE HANDLING

1. MODIFICATION TO TELOS
   PREDICTIVE AWAITIO HANDLING

2. DATA PASSING BY ADDRESS TO SAVE CPU CYCLES

3. IMMEDIATE REPLY ON MESSAGE RECEIPT

IPM MESSAGE HANDLING ALSO DESIGNED TO ALLOW INTERACTION WITH SYSTEMS USING TMF OR REQUIRING HSSPOOL TO BE IDENTIFIED AS A PRINT SPOOLER.

DISK PROCESSING

1. MODIFICATION TO THE QUEUE HANDLING
   CHAIN AND LINK CLASSES ALTERED TO ALLOW
   CONSTRUCTION OF MESSAGES FROM COMPONENTS

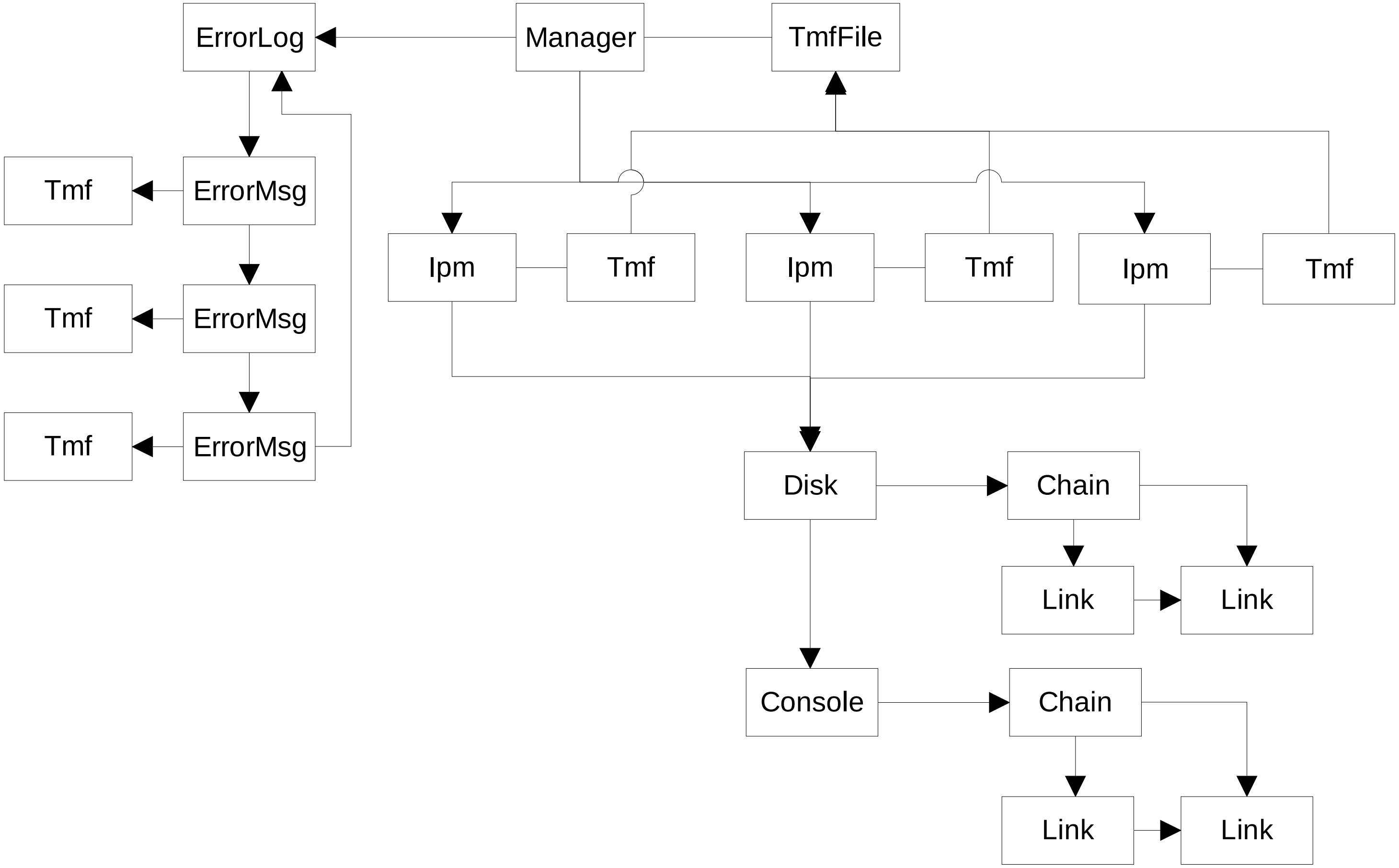
2. RECORD BLOCKING IN PROGRAM WITH
   WRITES TO DISK WHEN REQUIRED OR DURING
   PERIODS OF LOW ACTIVITY

MONITOR

1. SUSPENDED DURING PERIODS OF HIGH ACTIVITY

2. SINCE THE MONITOR CAN BE TEXT TOKENISED EMS. HSSPOOL CAN BE USED AS A SIMPLE, TEXT TO EMS MESSAGE CONVERTER

COLLECTOR STRUCTURE



HSSPOOL QUERY

NEED TO GET:

1. SUMMARY QUERIES – INQUIRY COMMAND
2. DETAILED QUERIES – PRINT COMMAND

FROM THE DATA STORE

FILTERED BY:

1. DATE/TIME
2. MESSAGE SOURCE
3. MESSAGE TYPE
4. MESSAGE CONTENTS

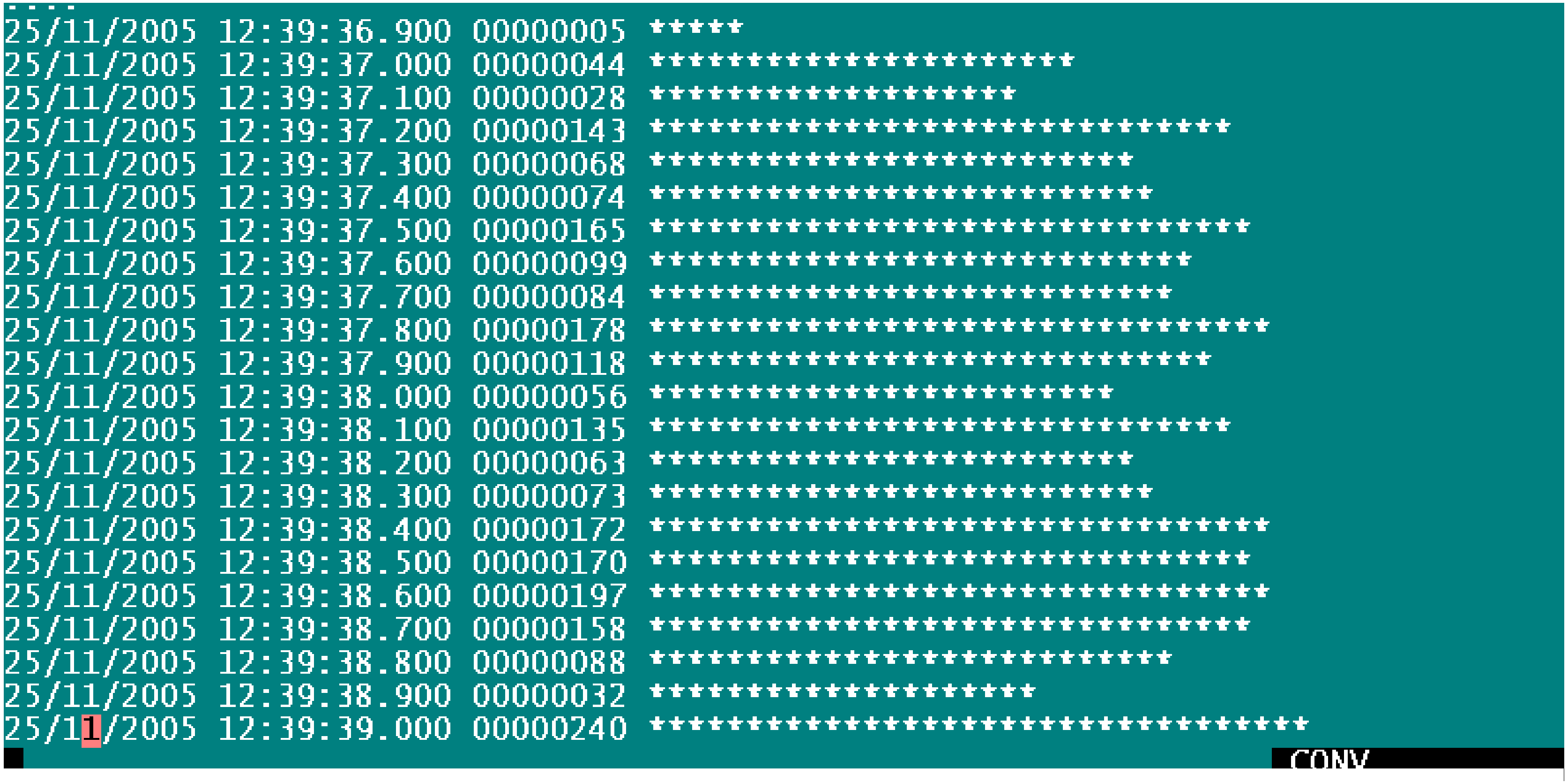
QUERY CONTROL

NEED TO DEFINE:

1. THE DATA SET ON WHICH THE QUERY IS TO BE PERFORMED

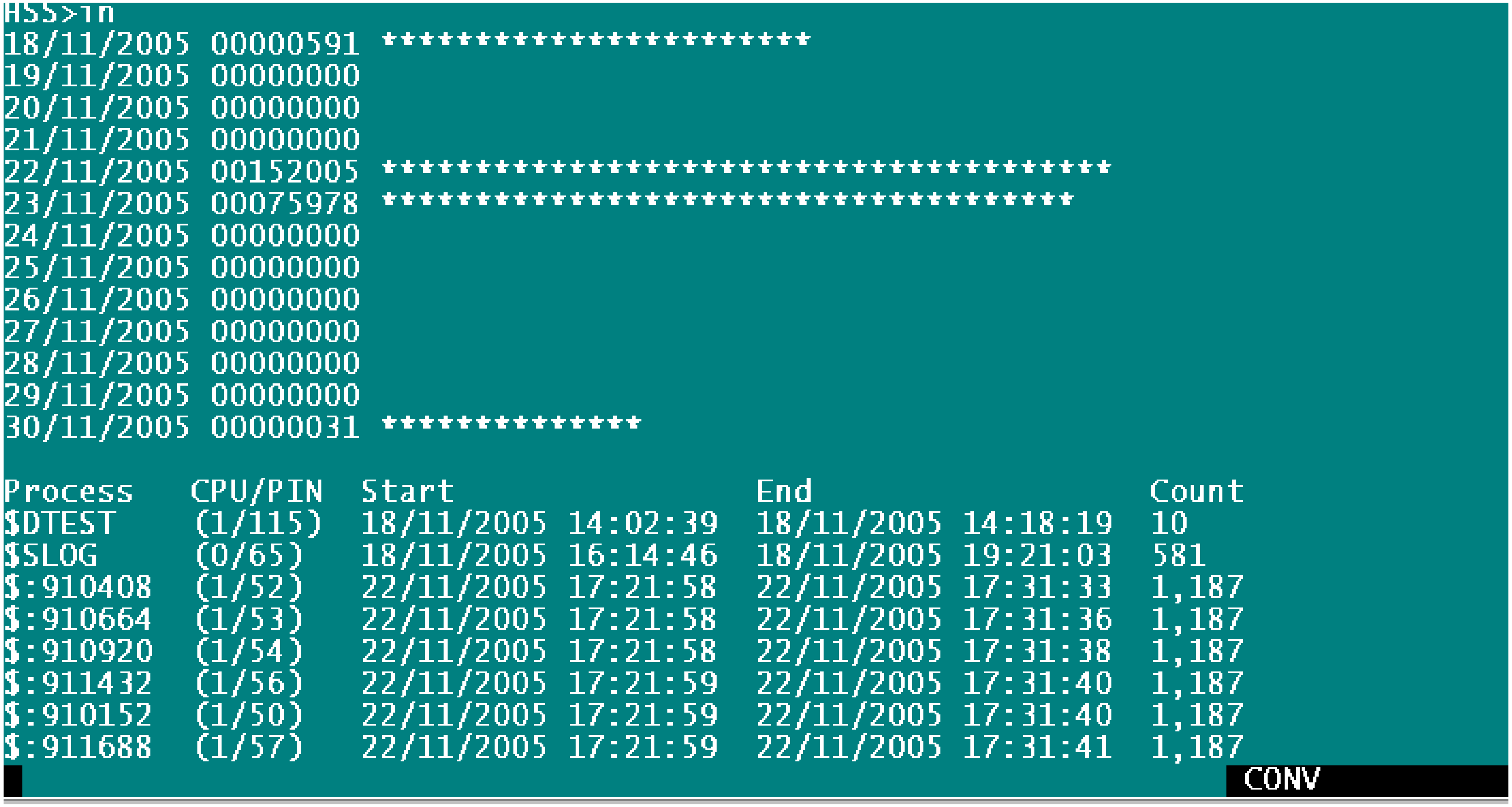
2. THE PRINT DEVICE OR SCRIPT FILE TO WHICH THE QUERY OUTPUT IS TO BE SENT

INQUIRY EXAMPLE 1 SEC.



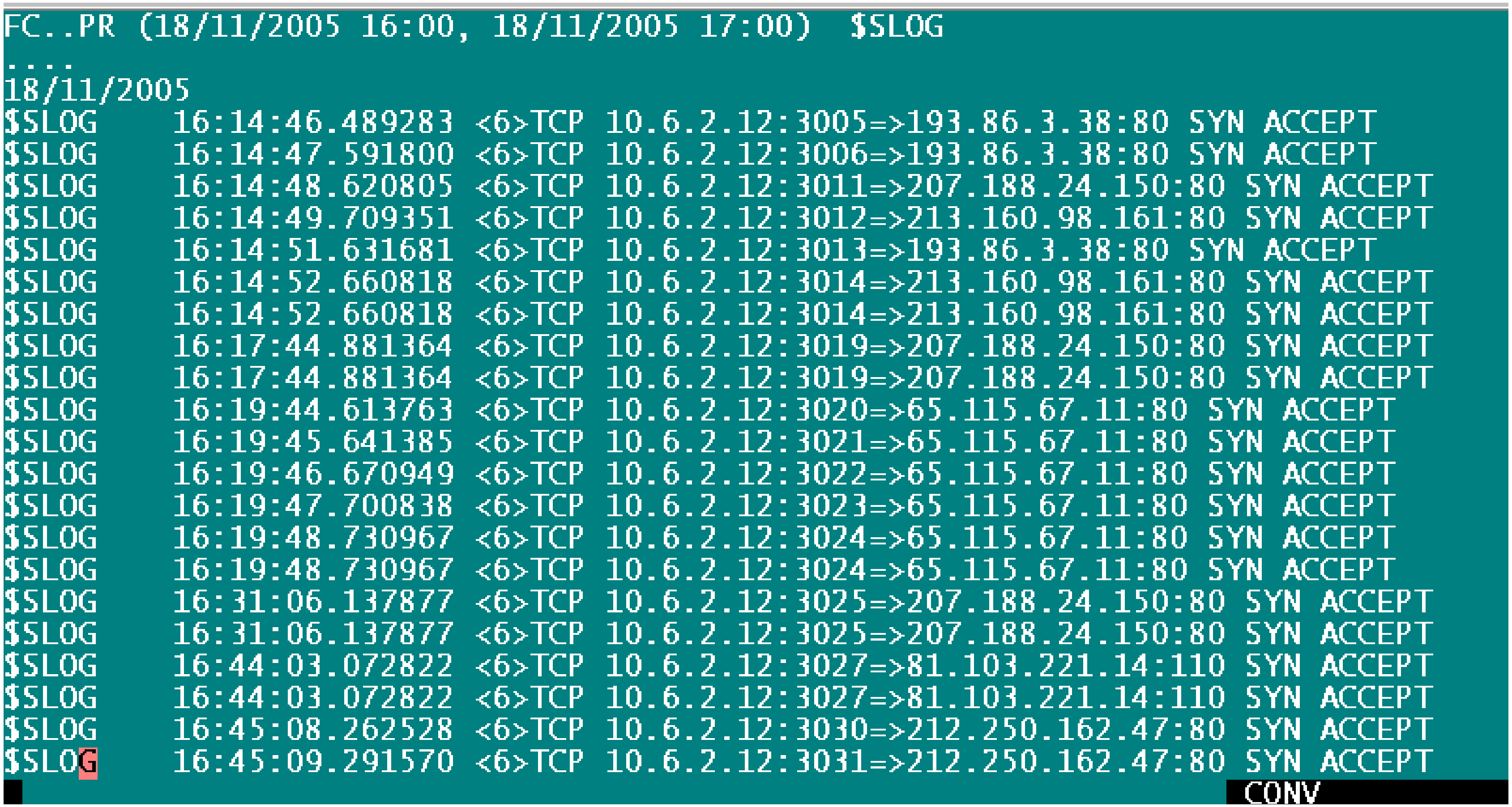
```
....
25/11/2005 12:39:36.900 00000005 *****
25/11/2005 12:39:37.000 00000044 **********************
25/11/2005 12:39:37.100 00000028 *******************
25/11/2005 12:39:37.200 00000143 ******************************
25/11/2005 12:39:37.300 00000068 *************************
25/11/2005 12:39:37.400 00000074 **************************
25/11/2005 12:39:37.500 00000165 *******************************
25/11/2005 12:39:37.600 00000099 ****************************
25/11/2005 12:39:37.700 00000084 ***************************
25/11/2005 12:39:37.800 00000178 ********************************
25/11/2005 12:39:37.900 00000118 *****************************
25/11/2005 12:39:38.000 00000056 ************************
25/11/2005 12:39:38.100 00000135 ******************************
25/11/2005 12:39:38.200 00000063 *************************
25/11/2005 12:39:38.300 00000073 **************************
25/11/2005 12:39:38.400 00000172 ********************************
25/11/2005 12:39:38.500 00000170 *******************************
25/11/2005 12:39:38.600 00000197 ********************************
25/11/2005 12:39:38.700 00000158 *******************************
25/11/2005 12:39:38.800 00000088 ***************************
25/11/2005 12:39:38.900 00000032 ********************
25/11/2005 12:39:39.000 00000240 **********************************
                                                                  CONV
```

INQUIRY EXAMPLE BY DAY



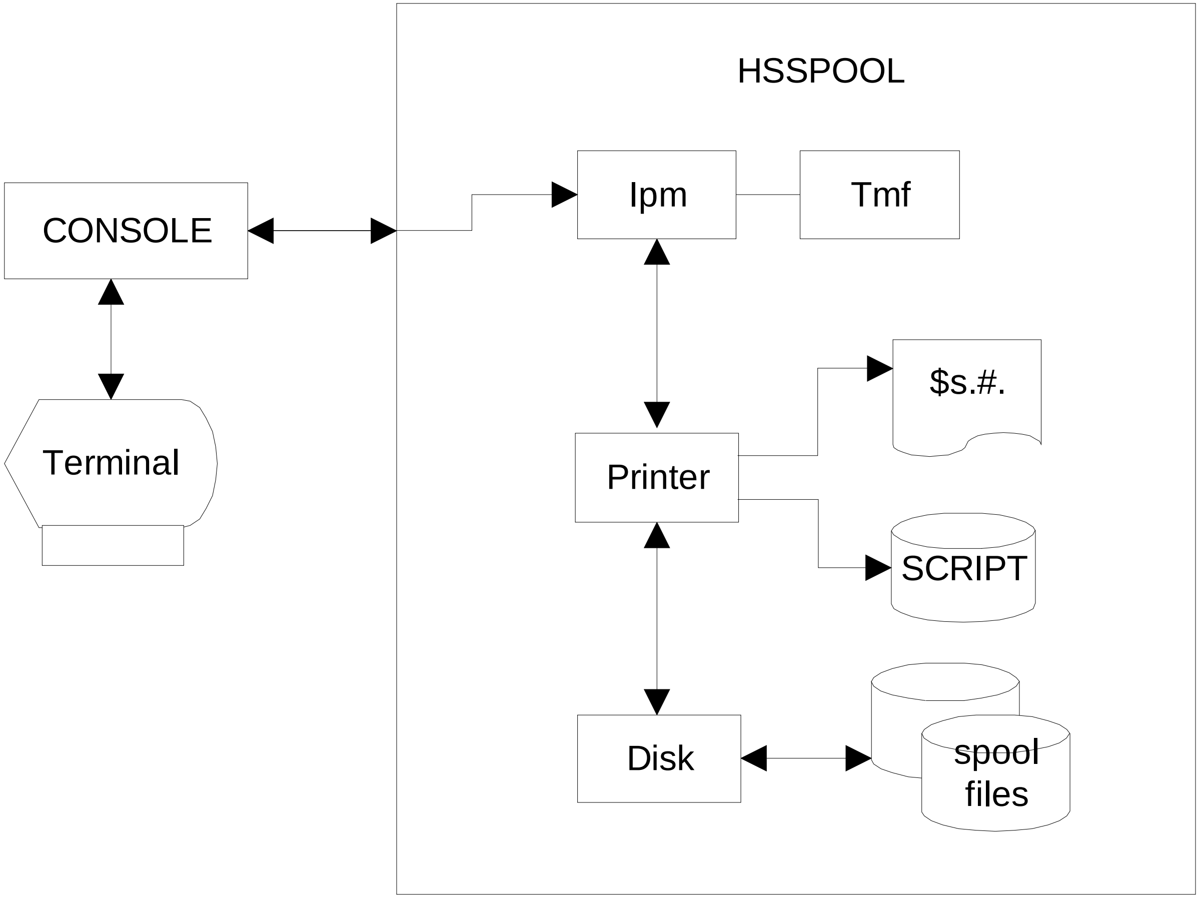
```
HSS>in
18/11/2005 00000591 ***********************
19/11/2005 00000000
20/11/2005 00000000
21/11/2005 00000000
22/11/2005 00152005 ***************************************
23/11/2005 00075978 *************************************
24/11/2005 00000000
25/11/2005 00000000
26/11/2005 00000000
27/11/2005 00000000
28/11/2005 00000000
29/11/2005 00000000
30/11/2005 00000031 **************

Process    CPU/PIN   Start                End                  Count
$DTEST      (1/115)  18/11/2005 14:02:39  18/11/2005 14:18:19  10
$SLOG       (0/65)   18/11/2005 16:14:46  18/11/2005 19:21:03  581
$:910408    (1/52)   22/11/2005 17:21:58  22/11/2005 17:31:33  1,187
$:910664    (1/53)   22/11/2005 17:21:58  22/11/2005 17:31:36  1,187
$:910920    (1/54)   22/11/2005 17:21:58  22/11/2005 17:31:38  1,187
$:911432    (1/56)   22/11/2005 17:21:59  22/11/2005 17:31:40  1,187
$:910152    (1/50)   22/11/2005 17:21:59  22/11/2005 17:31:40  1,187
$:911688    (1/57)   22/11/2005 17:21:59  22/11/2005 17:31:41  1,187
                                                               CONV
```

PRINT BY TIME & PROCESS



```
FC..PR (18/11/2005 16:00, 18/11/2005 17:00)  $SLOG
....
18/11/2005
$SLOG     16:14:46.489283 <6>TCP 10.6.2.12:3005=>193.86.3.38:80 SYN ACCEPT
$SLOG     16:14:47.591800 <6>TCP 10.6.2.12:3006=>193.86.3.38:80 SYN ACCEPT
$SLOG     16:14:48.620805 <6>TCP 10.6.2.12:3011=>207.188.24.150:80 SYN ACCEPT
$SLOG     16:14:49.709351 <6>TCP 10.6.2.12:3012=>213.160.98.161:80 SYN ACCEPT
$SLOG     16:14:51.631681 <6>TCP 10.6.2.12:3013=>193.86.3.38:80 SYN ACCEPT
$SLOG     16:14:52.660818 <6>TCP 10.6.2.12:3014=>213.160.98.161:80 SYN ACCEPT
$SLOG     16:14:52.660818 <6>TCP 10.6.2.12:3014=>213.160.98.161:80 SYN ACCEPT
$SLOG     16:17:44.881364 <6>TCP 10.6.2.12:3019=>207.188.24.150:80 SYN ACCEPT
$SLOG     16:17:44.881364 <6>TCP 10.6.2.12:3019=>207.188.24.150:80 SYN ACCEPT
$SLOG     16:19:44.613763 <6>TCP 10.6.2.12:3020=>65.115.67.11:80 SYN ACCEPT
$SLOG     16:19:45.641385 <6>TCP 10.6.2.12:3021=>65.115.67.11:80 SYN ACCEPT
$SLOG     16:19:46.670949 <6>TCP 10.6.2.12:3022=>65.115.67.11:80 SYN ACCEPT
$SLOG     16:19:47.700838 <6>TCP 10.6.2.12:3023=>65.115.67.11:80 SYN ACCEPT
$SLOG     16:19:48.730967 <6>TCP 10.6.2.12:3024=>65.115.67.11:80 SYN ACCEPT
$SLOG     16:19:48.730967 <6>TCP 10.6.2.12:3024=>65.115.67.11:80 SYN ACCEPT
$SLOG     16:31:06.137877 <6>TCP 10.6.2.12:3025=>207.188.24.150:80 SYN ACCEPT
$SLOG     16:31:06.137877 <6>TCP 10.6.2.12:3025=>207.188.24.150:80 SYN ACCEPT
$SLOG     16:44:03.072822 <6>TCP 10.6.2.12:3027=>81.103.221.14:110 SYN ACCEPT
$SLOG     16:44:03.072822 <6>TCP 10.6.2.12:3027=>81.103.221.14:110 SYN ACCEPT
$SLOG     16:45:08.262528 <6>TCP 10.6.2.12:3030=>212.250.162.47:80 SYN ACCEPT
$SLOG     16:45:09.291570 <6>TCP 10.6.2.12:3031=>212.250.162.47:80 SYN ACCEPT
                                                                   CONV
```

DISTRIBUTOR STRUCTURE



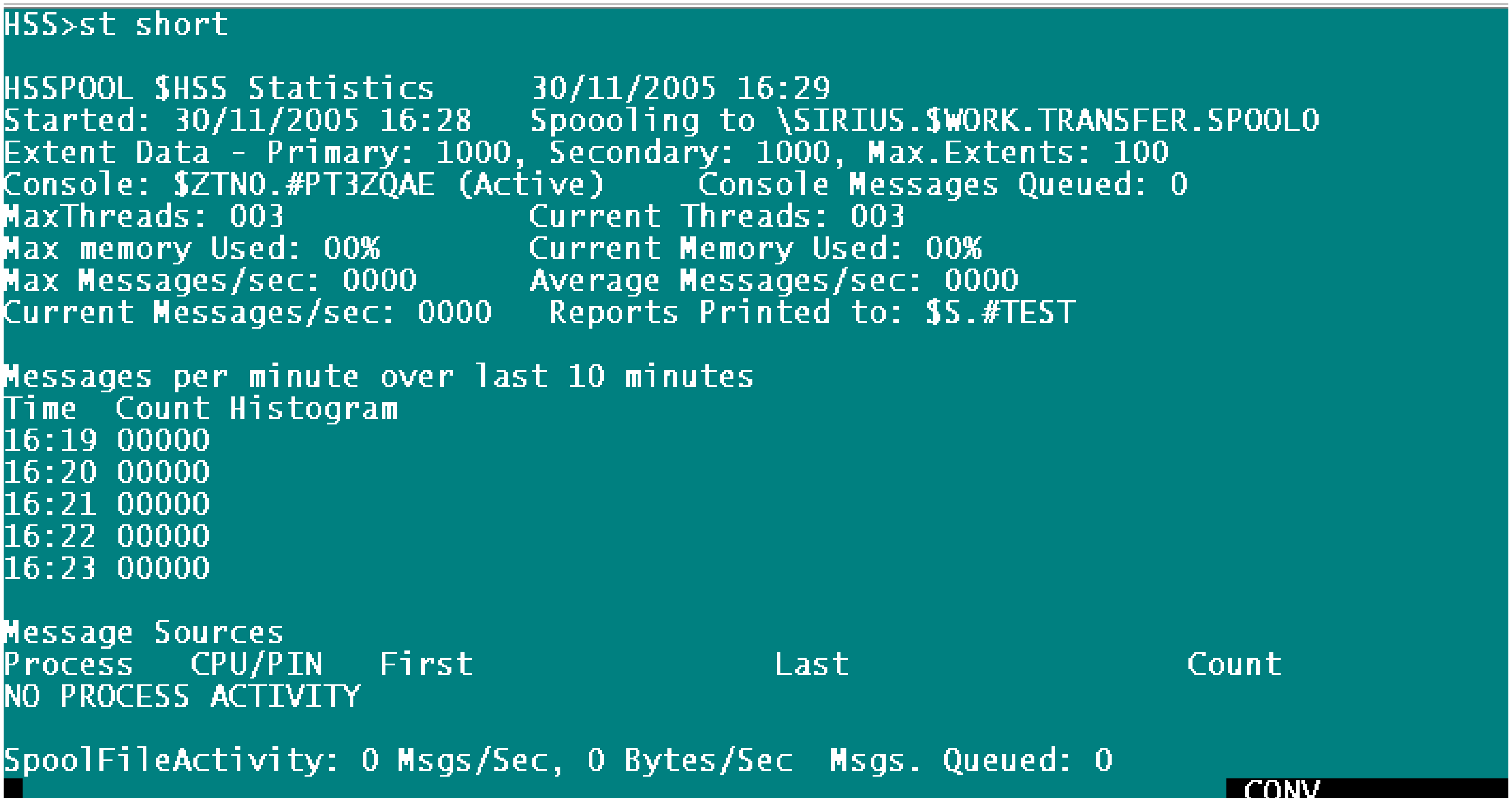
CONTROL COMMANDS

1. Spool — Define File Name and Extent Information,
   Swap Files

2. Monitor — Define Name (default self), Swap, Enable/Disable

3. Statistics — Get Performance Statistics

STATISTICS COMMAND



```
HSS>st short

HSSPOOL $HSS Statistics     30/11/2005 16:29
Started: 30/11/2005 16:28   Spoooling to \SIRIUS.$WORK.TRANSFER.SPOOL0
Extent Data - Primary: 1000, Secondary: 1000, Max.Extents: 100
Console: $ZTN0.#PT3ZQAE (Active)     Console Messages Queued: 0
MaxThreads: 003             Current Threads: 003
Max memory Used: 00%        Current Memory Used: 00%
Max Messages/sec: 0000      Average Messages/sec: 0000
Current Messages/sec: 0000   Reports Printed to: $S.#TEST

Messages per minute over last 10 minutes
Time  Count Histogram
16:19 00000
16:20 00000
16:21 00000
16:22 00000
16:23 00000

Message Sources
Process   CPU/PIN   First                    Last                  Count
NO PROCESS ACTIVITY

SpoolFileActivity: 0 Msgs/Sec, 0 Bytes/Sec  Msgs. Queued: 0
                                                                   CONV
```

HSSPOOL SUMMARY

1. Capture speed. Up to 2400 messages per second on S Series
2. No need to modify your code. Capture process available
3. Trace library available.
4. Simple insertion of application tracing code
5. Text Message, Binary Dumps and Capture Messages processing
6. Control Interface, Comprehensive and Intuitive
7. Fast analysis by Process, Date/Time, Message Type, Contents
8. Summary and Detailed Reports to Printer or Edit Files
9. Capture Message Script Files can be generated for replay

QUESTIONS

IF YOU WANT TO CONTACT US

Tel.	+44(0)1206-392923
e-Mail:	info@rsi-ns.com