

TO: Distribution
FROM: Jay Goldman
DATE: 08/11/74
SUBJECT: A generalized ready message program

Attached to this MTB is the description of a generalized ready message command. It is a modification of the author maintained `general_ready` command. This new version is capable of printing both incremental and total values in the same ready message (although the total values are per-process totals only). The date and time can be printed in a number of different formats and new 6180 Multics quantities such as memory units, level numbers, etc. can also be printed.

The new version of `general_ready` described in the attached MPM like description is operational, a version is available in `>udc>RDMS>public>general_ready`.

One addition that has already been suggested is the replacement of the print messages facility with the more general capability of specifying a command line to be executed whenever the ready message procedure is invoked.

Additional comments or suggestions should be directed to Jay Goldman MIT room 39-411 or to Goldman.RDMS on the MIT Multics.

```

|-----|
| general_ready |
|-----|

```

MULTICS PROGRAMERS' MANUAL

Author-Maintained Library
 Jay Goldman
 39-411
 Ext. 3-4107
 08/11/74

Name: general_ready, gr

general_ready is a command which allows the user to set up a ready message containing specific values in a user specified format. The program is designed to allow an almost arbitrary format at no additional cost (relative to the system's ready procedure) other than the cost associated with the general_ready command which sets up the ready message. Once a ready message has been specified, the ready_on, ready, and ready_off commands will control the printing of the ready message in the normal manner.

general_ready builds up an loa_ control string from the order of the keywords passed to it. The keywords specify which values to output in the ready message. Virtual cpu usage, real cpu usage, and dollar cost can be printed. Both incremental usage (usage accrued since the last ready message printed by general_ready) and total usage (usage accrued during this process) can be printed in the same ready message with the precision of the output (the number of decimal places to the right of the decimal point) specified by the user.

Usage

```

general_ready      -control options-      -prefix options-
-format options-

```

The prefix options affect the format of the ready message but do not specify the contents (the values printed) of the ready message. The format options specify both the contents and the format of the ready message. The control options do not affect the format of the ready message, but instead control how the ready message and other facilities provided by general_ready are used. The prefix options must occur before any format options; while the control options may appear anywhere on the command line.

Format Options

The format and content of the ready message is controlled by format options. These options include: keywords which identify values to be included in the ready message; optional arguments

```

-----
| general_ready |
-----

```

following these keywords which control the number of digits after the decimal point in numeric values; and literal character strings which are inserted in the ready message. The format options are combined in the order of their appearance in the `general_ready` command to form an `ioa_` control string which controls the format of the ready message.

Six types of values may be used in a ready message: processor usage values (virtual and real cpu seconds); memory usage values (memory units); paging operations (both bulk store reads and demand page faults); usage cost values (dollar charges); command processor level numbers; and date/time values (date, time of day, day of the week, etc). Both total usage values (total usage accrued during this process) and incremental usage values (usage accrued since the last ready message printed by `general_ready`) can be output in the same ready message. The values are selected for use in the ready message by the format keywords which are given in the command. The format keywords are listed below by type.

- 1) processor usage values are selected from the following keywords. These keywords can be followed by an optional argument, a single numeric digit from 1 to 9, to indicate the number of digits which should appear to the right of the decimal point in the number which is output. The default is 3 digits. The output format of the value can be described by the `ioa_` control string "`~.p`" where `p` is 3 by default.

```

-ri
-rcpui incremental real cpu value

```

```

-rt
-rcput total real cpu value

```

```

-vi
-vcpui incremental virtual cpu value

```

```

-vt
-vcput total virtual cpu value

```

- 2) memory usage values are selected by the following keywords. These keywords are used in the same manner as the keywords for processor usage values above.

```

-memoryi
-mui

```

```

|-----|
| general_ready |
|-----|

```

Page 4

-memi Incremental memory units

-memoryt

-memt

-mut total memory units

- 3) Usage cost values are selected by the following keywords. These keywords are used in the same manner as the keywords for processor usage values with the following differences. The default number of digits following the decimal point is 2. The output format of the value can be described by the loa_ control string "\$-p.f" where p is 2 by default.

-dollarsi

-\$i Incremental cost charges

-dollarst

-\$t total cost charges

- 4) paging values are selected from the following keywords. These keywords are output by the loa_ control string "~d+~d", where the first number is the number of bulk store pages read (formerly the number of pre-pages) and the second is the number of demand page faults.

-pagesi

-pgi incremental paging values

-pagest

-pgt total paging values

- 5) command processor level numbers are selected by the following keyword. This keyword indicates that the command processor and stack frame level numbers should be included in the ready message. No optional argument may be used to control the number of digits. The level numbers are output by the loa_ control string "~a", but the printed format can be described by "level ~d,~d" where the first number is the number of command processor invocations and the second is the stack frame depth of the ready message procedure's stack frame. If the command processor level is 1, the printed format is the null string.

-level

-lv command processor level numbers

```

-----
| general_ready |
-----

```

6) date values are selected by the following keywords. These values can be described by the loa control string "~a" except for the -min keyword which uses the loa control string "~a" (without a leading space). No optional argument may be used to control the number of digits.

```

-dt      date and time (mm/dd/yy hhmm.m zzz www)

-date    8 character date (mm/dd/yy)

-hour    2 digit hour (hh)

-minute
-min     2 digit minute (mm)

-time
-tm      6 digit time of day (hhmm.m)

-dow     3 character day of week (www)

-zone    3 character time zone (zzz)

```

Note that all values except for the number of minutes (-minute) are preceded by a space and none of the values are suffixed by a space. Any non-keyword argument (other than a single numeric digit following a floating point or dollar keyword) is assumed to be a literal string which is inserted in loa_control string being built by general_ready. Refer to the examples below.

Control Options

The following control options affect the operation of general_ready, but do not change the format of ready messages.

```

-set     causes general_ready to establish itself as the current
         ready message procedure. The command processor will
         then call general_ready to print a ready message after
         each command line is complete. In addition, the system
         commands, ready, ready_on, and ready_off, will affect
         the operation of general_ready.

-reset   causes general_ready to make the system ready proc the
-rs      current ready message procedure, and to reset any timer
         alarms which were established to catch shift changes.

```



```

-----
| general_ready |
-----

```

message. The `ioa_control` string formats for the various values which can be inserted into the ready message are given above for each type of value. This argument overrides any format control options which would normally affect the format of the ready message. However, format keywords must still be specified to indicate which values are to be output and the order in which these values correspond with the `ioa_control` characters in the control string.

Examples

The following examples illustrate some of `general_ready`'s facilities:

```
gr -string READY -date ^xTIME -time ^xVCPU -vl -vcput -set
```

The above command line establishes `general_ready` as the current ready procedure since the `-set` keyword appeared. Each ready message would have the format:

```
READY 01/15/74 TIME 1234.3 VCPU 3.456 23.349
```

If the `-set` keyword had not appeared, a single ready message having the above format would be printed.

The `ioa_control` string which `general_ready` uses to generate the above ready message would be:

```
"READY ^a^xTIME ^a^xVCPU ^f^f^2/"
```

The command line:

```
gr -s READY -date -hour : -min ^xVCPUI -vcput ^xVCPUI  
-vcput 2
```

would result in a single ready message of the form:

```
READY 01/15/74 09:46 VCPUI 2.345 VCPUI 34.21
```

using the `ioa_control` string:

```
"READY ^a ^a:^a^xVCPUI ^f^f^xVCPUI ^f^f^2/"
```

The above ready message could have been specified by the command

```
-----  
| general_ready |  
|-----|
```

Page 8

line:

```
gr -ctl "READY ^a ^a:^a VCPUI ^3f VCPUT ^2f^2/" -date  
-hour -min -vcgui -vcput
```

Note

The total values printed by `general_ready` are per-process totals only. Since the system does not provide usage information for processes other than the current process, `general_ready` can not obtain information about usage in prior processes. When `general_ready` is invoked for the first time in a process, the dollar cost of all usage (in that process) up to that point in time is computed at the rates then in effect.