Guy Harris

THE MULTICS QUIT HANDLER

The following graphs display the path followed by the Multics system in processing a quit. In the first graph, the vertical distance is proportional to the number of calls and returns while in the second graph it is proportional to the estimated number of instructions executed. The horizontal distance is proportional to the stack depth.

This trace assumes that
1) no links need to be snapped
2) no segment or page faults occur
3) no timer runouts occur.

The trace begins with the interrupt from the attention signal. We assume that
4) the terminal is a modified 2741.
5) the terminal was in read mode at the time.
The typewriter DIM frees the read chain, sends the target process a "quit" interprocess signal, starts the channel with sending a NL, and returns from the interrupt.

When the typewriter is finished, the GIOC again interrupts the CPU. The typewriter DIM then prepares and starts the read chain, and returns from the interrupt.

After the keyboard addressing sequence of the read chain is complete, another interrupt occurrs. The typewriter DIM frees the block used for addressing the keyboard, and returns.

When the traffic controller schedules the target process, a process interrupt is caused. This interrupt passes through the FIM and signaller to signal_, which searches the ring 4 stack for the "quit" condition handler. We assume that
6) no such handler exists
so that the default handler for this ring is called. We further assume that
7) standard_default_handler_ is the default handler.
standard_default_handler_ calls ioa_ in order to print the "QUIT" message. ioa_ calls formline_ to format the message and then calls ios_$write to print it. We assume that
8) the standard i/o assignments are in affect.
ios_ calls ttydim$tty_write, which then calls hcs_$tty_write. This call passes through the FIM and the gatekeeper, and then goes to tty_write. tty_write calls tty_inter$prescan in order to find the current print position, and then calls tty_inter$write to start the write. tty_inter$write stops the channel and connects the active write chain. It then returns, and control passes all the way back to standard_default_handler_. standard_default_handler_ then calls get_to_cl$unclaimed_signal in order to return to command level. get_to_cl establishes the default handler, saves the current i/o attachments to user_input, user_output, and error_output, and establishes the default. It calls ios_$changemode on user_i/o to reestablish the default modes, and then calls listen_$release_stack in order to read the next command. listen_$release_stack establishes

a cleanup procedure, and calls cu_$ready_proc in order to print the ready message. We assume that

9) print_ready_message_ is the ready procedure.

print_ready_message_ calls ioa_$nnl in order to print the message. ioa_$nnl, in turn, calls formline_ to format the message and ios_$write_ptr to print it. This call passes through the typewriter DIM like previous calls, and returns to listen_$release_stack. listen_$release_stack then calls hcs_$reset working set in order to reset the working set data. We assume that

10) only a few pages need their used flags reset.

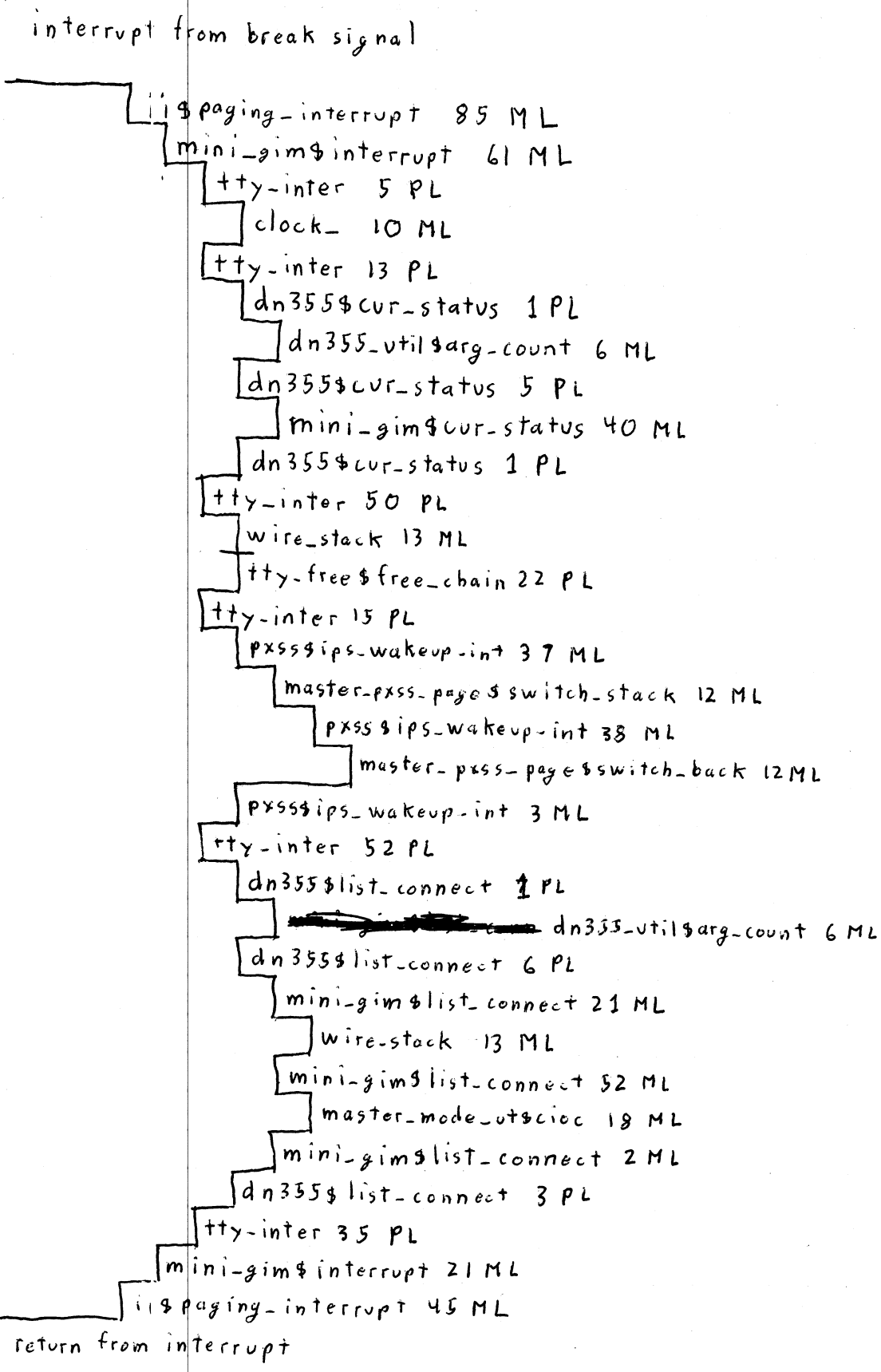
hcs_$reset_working_set returns to listen_$release_stack, which then calls ios_$read_ptr to read the command line. ios_$read_ptr calls ttydim_$tw_read, which calls in turn hcs_$tty_read. We assume that

11) no type-ahead has been performed

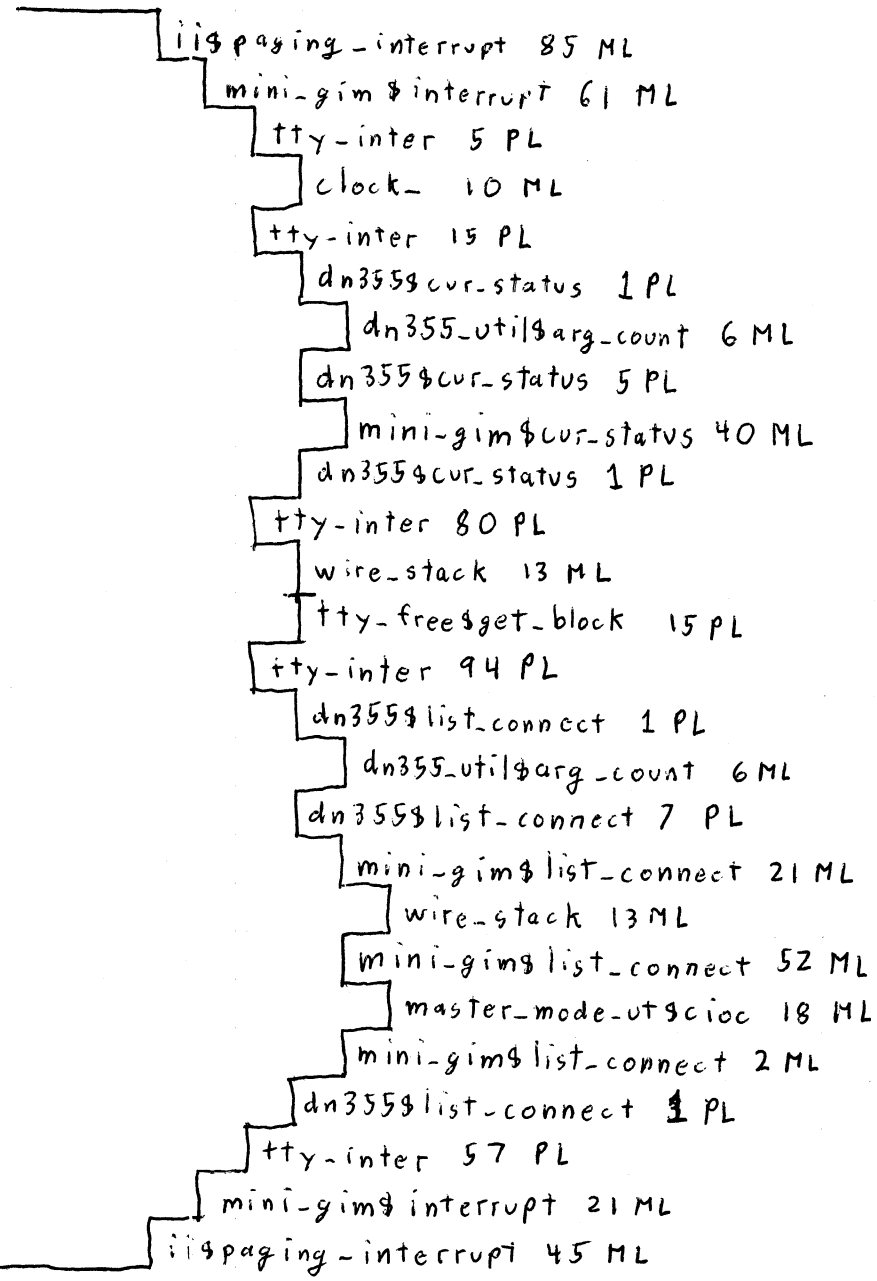
so that hcs_$tty_read retruns with an indication that no data is available. ttydim_$tw_read then calls ipc_$block to wait for input, and ips_$block calls hcs_$ipc_fblock to go blocked on the "special" typewriter event channel. hcs_$ipc_fblock then calls pxss$block in order to give up the processor.

After the printer has been addressed in order to print "QUIT", an interrupt occurs. The typewriter DIM frees the block used to address the printer and returns. The next interrupt occurs when the first (and only) block of the active write chain is done; the typewriter DIM frees the block and returns. Immediately afterwards, the interrupt signalling the end of the active chain occurs and the DIM swaps the write chains. Another end-of-block interrupt occurs when the first (and only) block of the new active write chain is done, and another end-of-chain interrupt occurs immediately afterwards. This time, the DIM prepares and connects the read chain. Finally, an interrupt is taken when the keyboard is addressed; the DIM then frees the block used to address the keyboard.

N.B. timeouts are assumed not to occur.

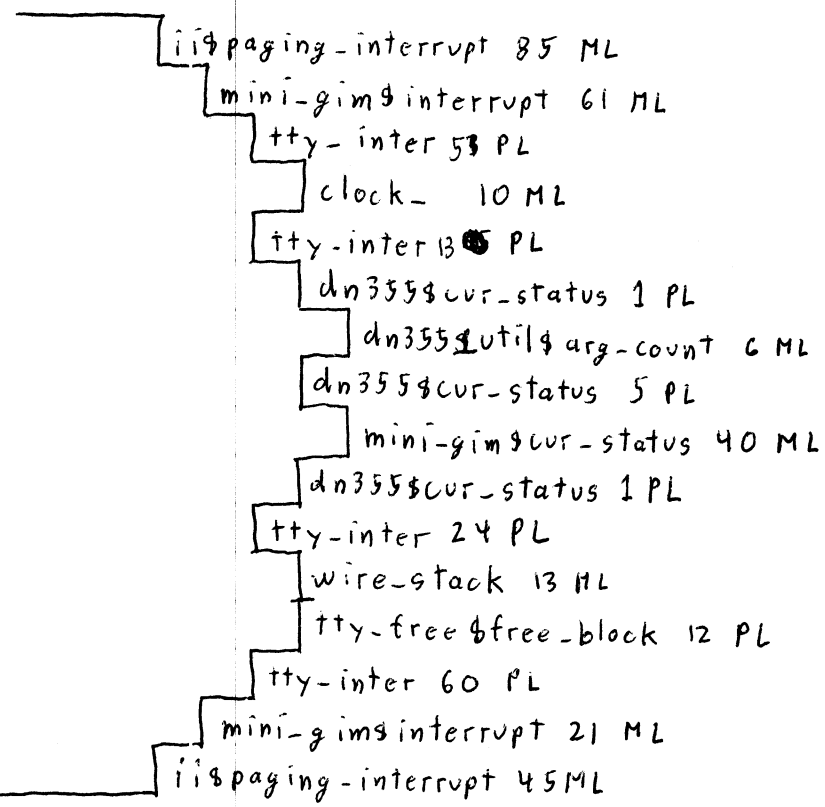

interrupt from break signal
i1$paging_interrupt 85 ML
mini_gim$interrupt 61 ML
tty-inter 5 PL
clock_ 10 ML
tty-inter 13 PL
dn355$cur_status 1 PL
dn355_util$arg_count 6 ML
dn355$cur_status 5 PL
mini_gim$cur_status 40 ML
dn355$cur_status 1 PL
tty-inter 50 PL
wire_stack 13 ML
tty-free$free_chain 22 PL
tty-inter 15 PL
pxss$ips_wakeup_int 37 ML
master_pxss_page$switch_stack 12 ML
pxss$ips_wakeup_int 38 ML
master_pxss_page$switch_back 12 ML
pxss$ips_wakeup_int 3 ML
tty-inter 52 PL
dn355$list_connect 1 PL
dn355_util$arg_count 6 ML
dn355$list_connect 6 PL
mini_gim$list_connect 21 ML
wire_stack 13 ML
mini_gim$list_connect 52 ML
master_mode_ut$cioc 18 ML
mini_gim$list_connect 2 ML
dn355$list_connect 3 PL
tty-inter 35 PL
mini_gim$interrupt 21 ML
i1$paging_interrupt 45 ML
return from interrupt

interrupt after printing newline



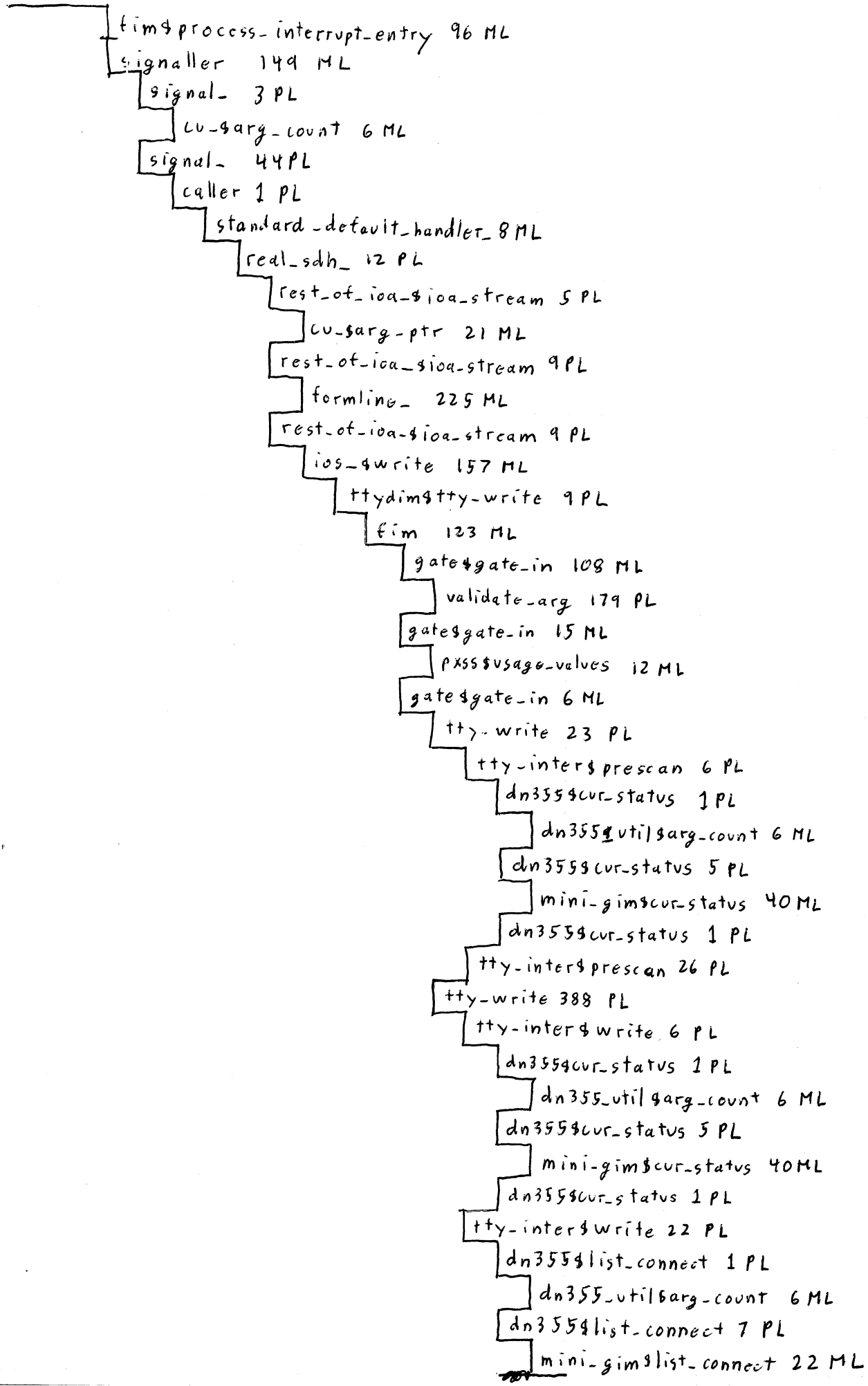
return from interrupt

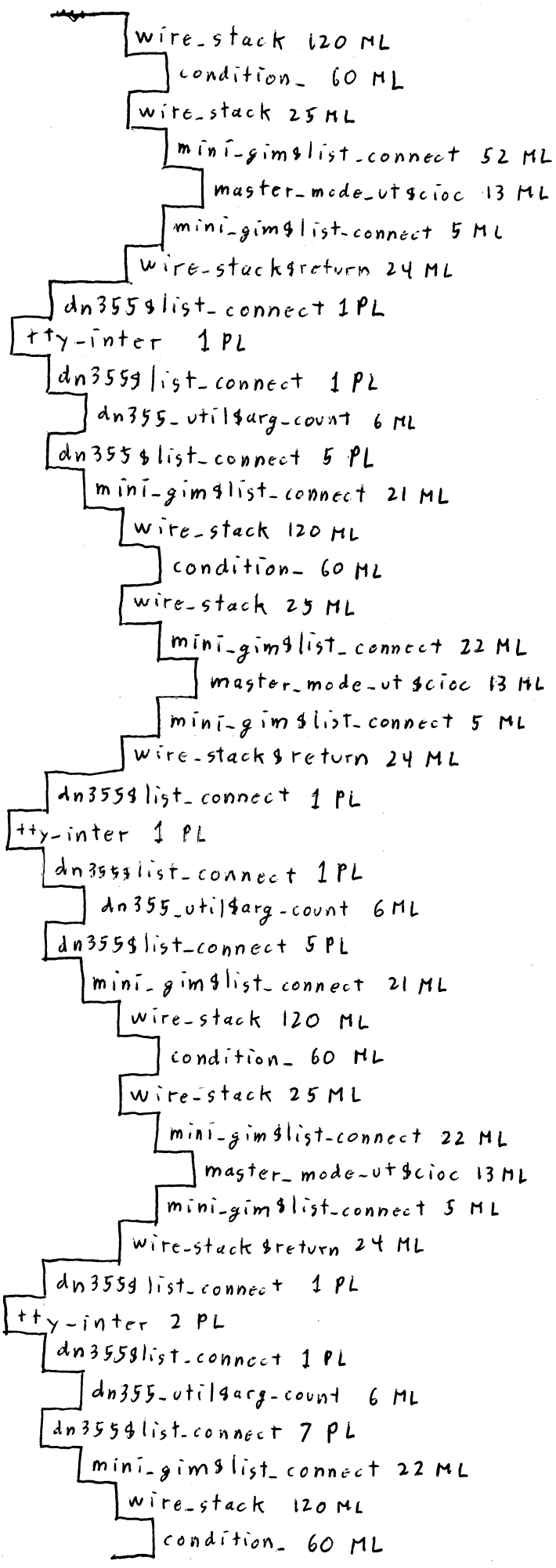
interrupt after addressing keyboard

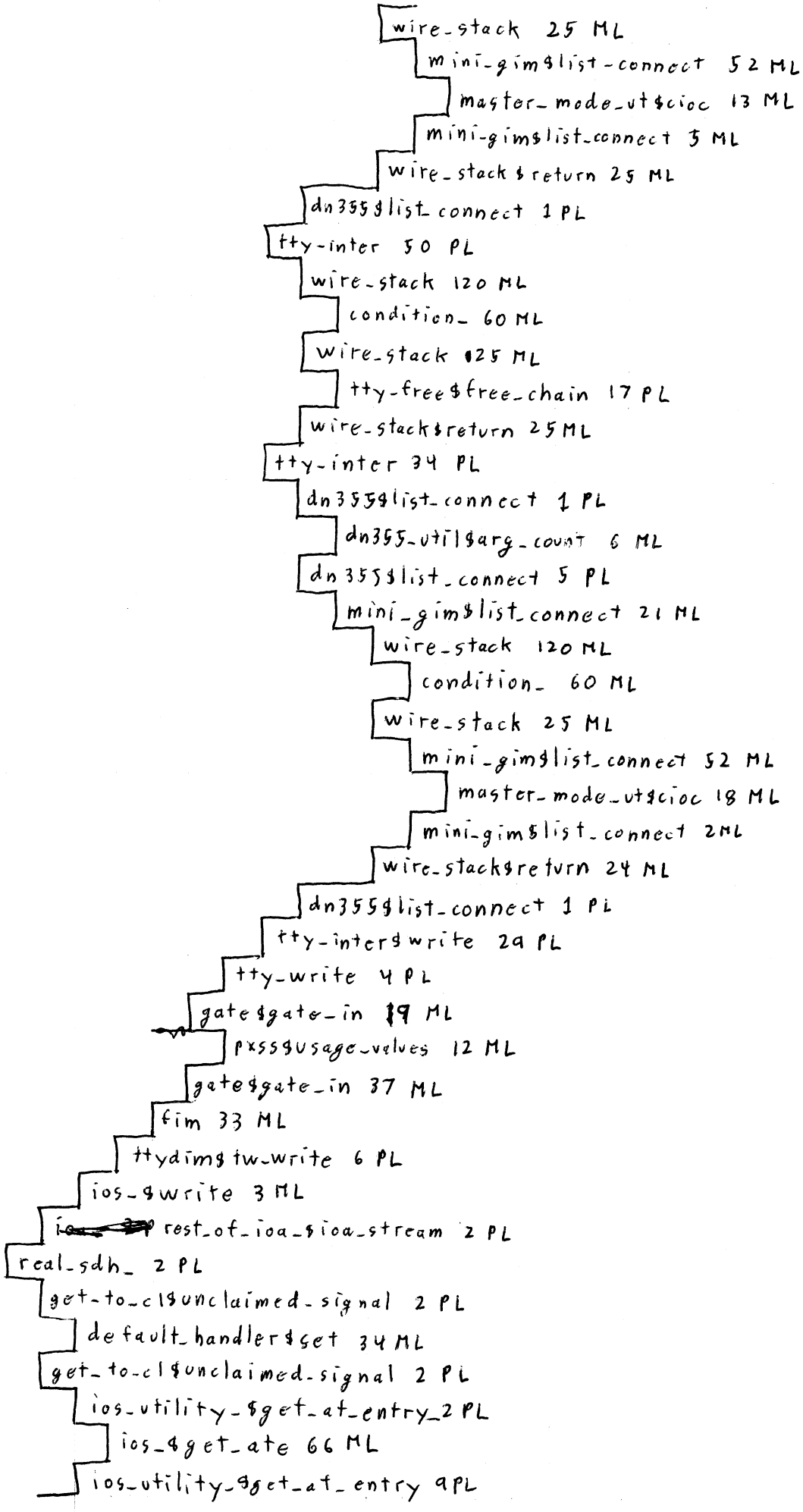


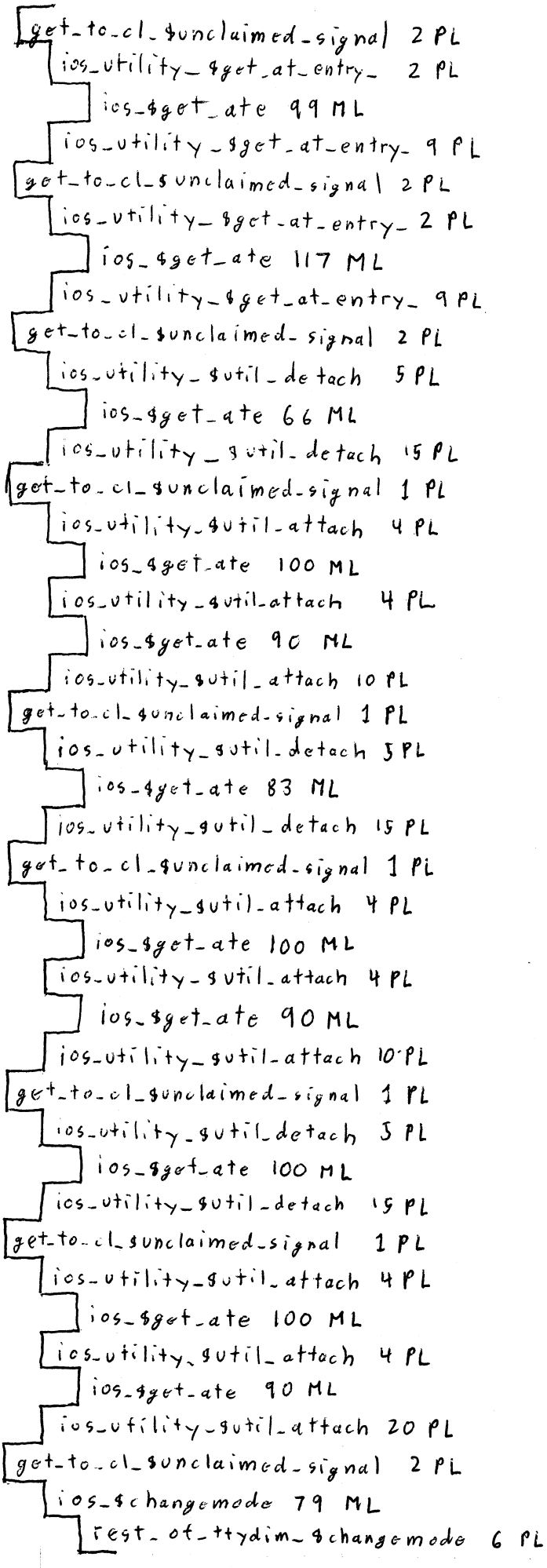
return to interrupted program

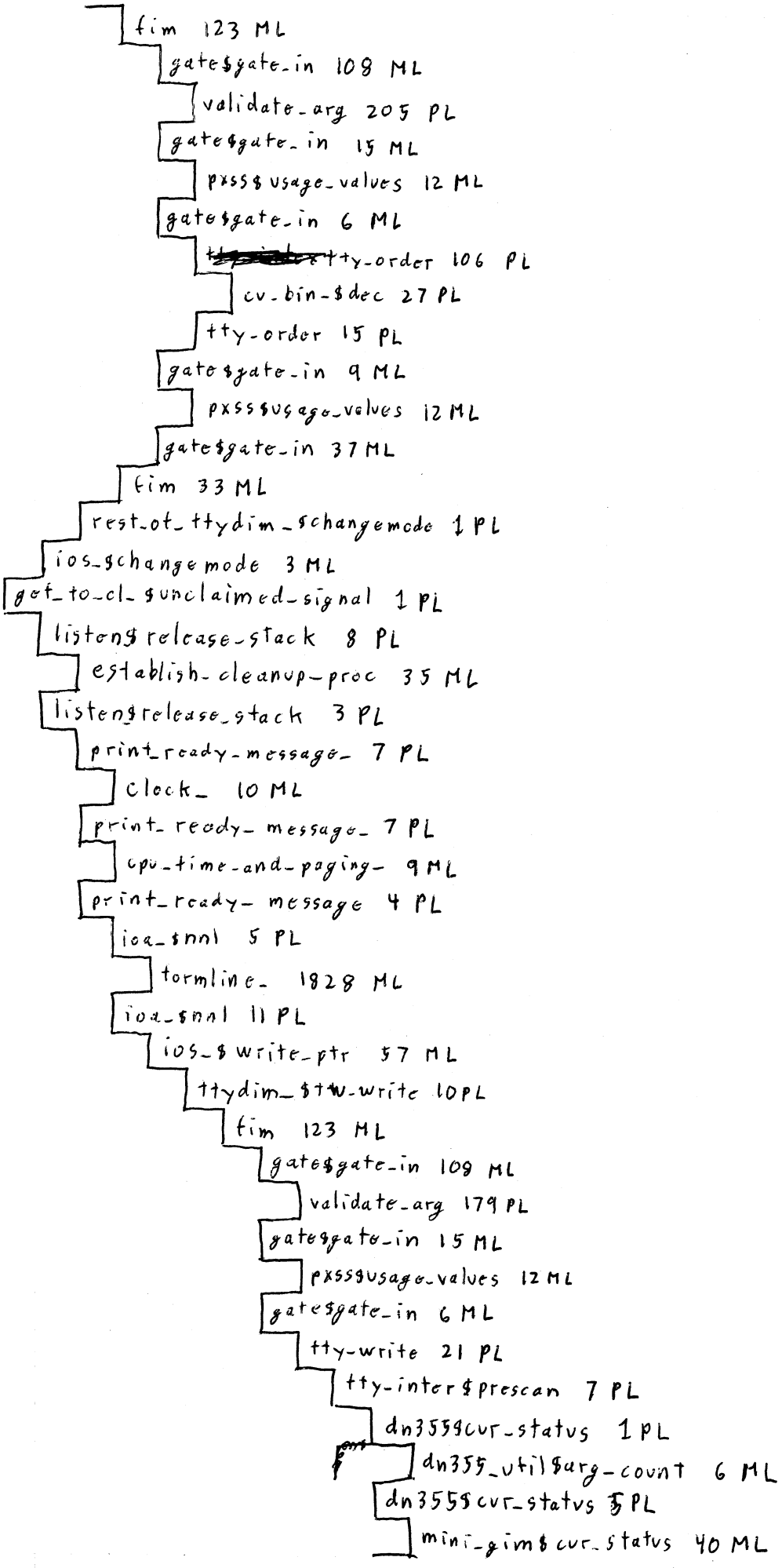
process interrupt (quit)

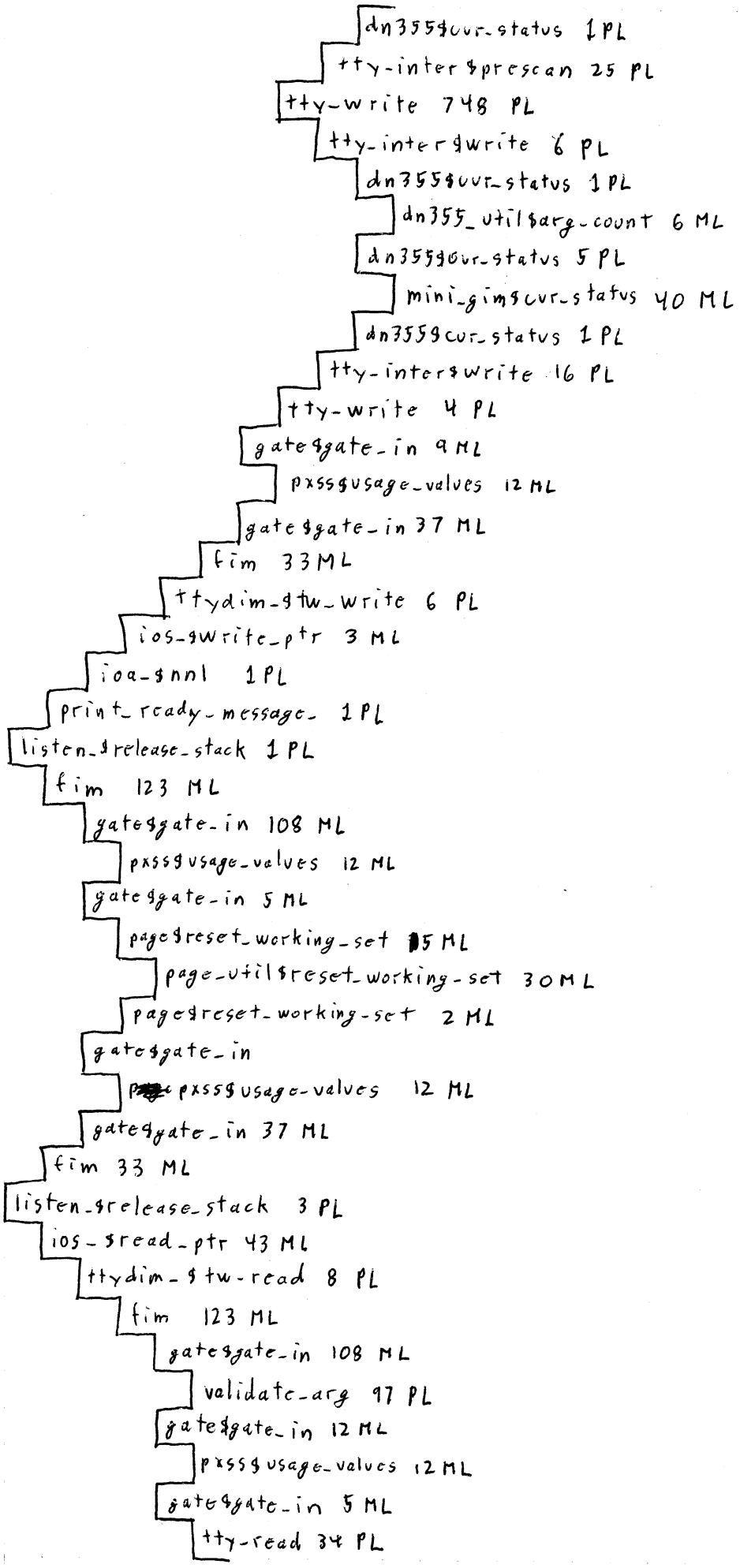


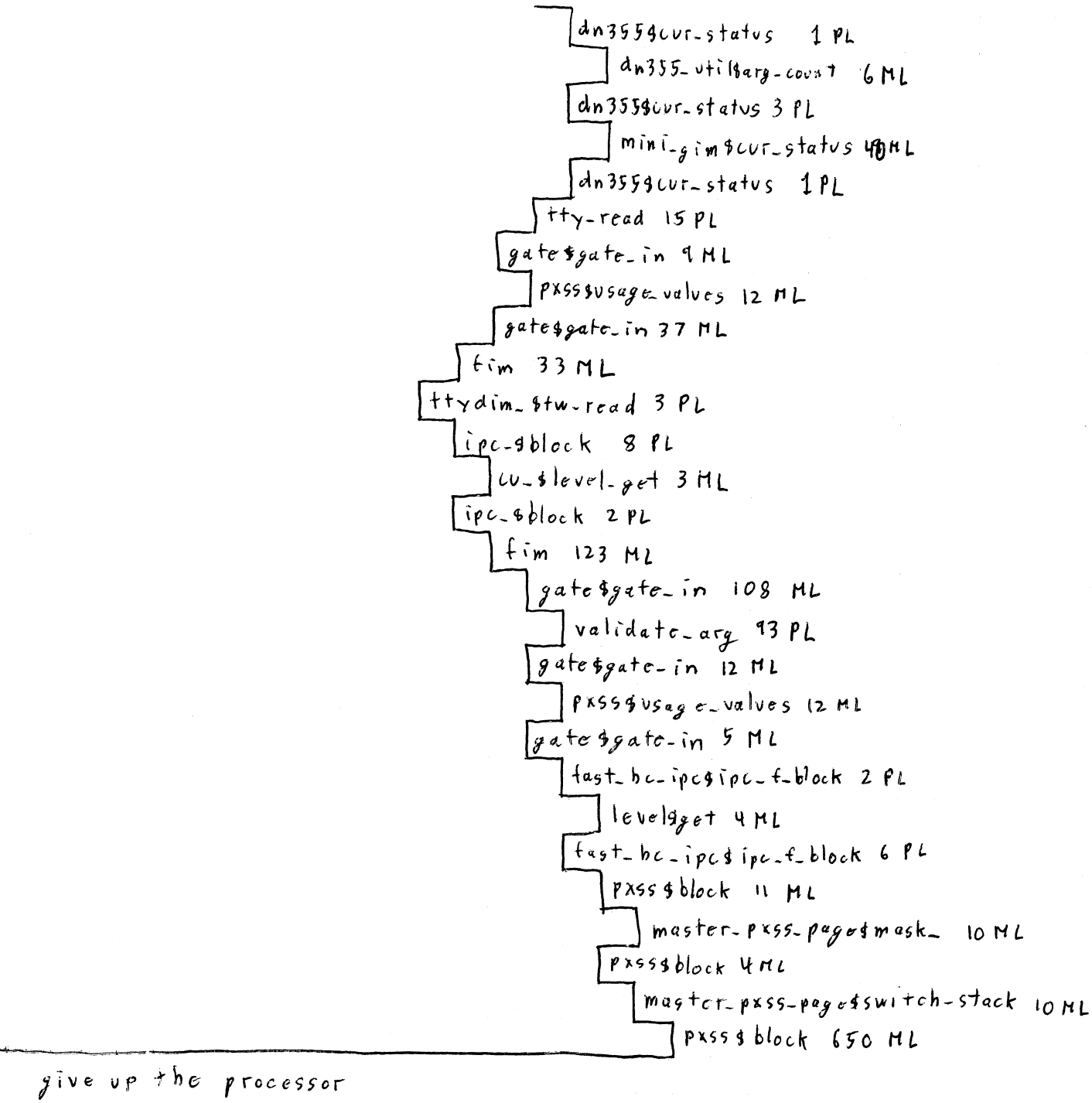

wire_stack 120 ML
condition_ 60 ML
wire_stack 25 ML
mini_gim$list_connect 52 ML
master_mode_ut$cioc 13 ML
mini_gim$list_connect 5 ML
wire_stack$return 24 ML
dn355$list_connect 1 PL
tty_inter 1 PL
dn355$list_connect 1 PL
dn355_util$arg_count 6 ML
dn355$list_connect 5 PL
mini_gim$list_connect 21 ML
wire_stack 120 ML
condition_ 60 ML
wire_stack 25 ML
mini_gim$list_connect 22 ML
master_mode_ut$cioc 13 ML
mini_gim$list_connect 5 ML
wire_stack$return 24 ML
dn355$list_connect 1 PL
tty_inter 1 PL
dn355$list_connect 1 PL
dn355_util$arg_count 6 ML
dn355$list_connect 5 PL
mini_gim$list_connect 21 ML
wire_stack 120 ML
condition_ 60 ML
wire_stack 25 ML
mini_gim$list_connect 22 ML
master_mode_ut$cioc 13 ML
mini_gim$list_connect 5 ML
wire_stack$return 24 ML
dn355$list_connect 1 PL
tty_inter 2 PL
dn355$list_connect 1 PL
dn355_util$arg_count 6 ML
dn355$list_connect 7 PL
mini_gim$list_connect 22 ML
wire_stack 120 ML
condition_ 60 ML


wire_stack 25 ML
mini_gim$list_connect 52 ML
master_mode_ut$cioc 13 ML
mini_gim$list_connect 3 ML
wire_stack$return 25 ML
dn355$list_connect 1 PL
tty_inter 50 PL
wire_stack 120 ML
condition_ 60 ML
wire_stack 025 ML
tty_free$free_chain 17 PL
wire_stack$return 25 ML
tty_inter 34 PL
dn355$list_connect 1 PL
dn355_util$arg_count 6 ML
dn355$list_connect 5 PL
mini_gim$list_connect 21 ML
wire_stack 120 ML
condition_ 60 ML
wire_stack 25 ML
mini_gim$list_connect 52 ML
master_mode_ut$cioc 18 ML
mini_gim$list_connect 2ML
wire_stack$return 24 ML
dn355$list_connect 1 PL
tty_inter$write 29 PL
tty_write 4 PL
gate$gate_in 19 ML
pxss$usage_velves 12 ML
gate$gate_in 37 ML
fim 33 ML
ttydim$tw_write 6 PL
ios_$write 3 ML
rest_of_ioa_$ioa_stream 2 PL
real_sdh_ 2 PL
get_to_cl$unclaimed_signal 2 PL
default_handler$set 34 ML
get_to_cl$unclaimed_signal 2 PL
ios_utility_$get_at_entry 2 PL
ios_$get_ate 66 ML
ios_utility_$get_at_entry 9PL


get_to_cl_$unclaimed_signal 2 PL
ios_utility_$get_at_entry_ 2 PL
ios_$get_ate 99 ML
ios_utility_$get_at_entry_ 9 PL
get_to_cl_$unclaimed_signal 2 PL
ios_utility_$get_at_entry_ 2 PL
ios_$get_ate 117 ML
ios_utility_$get_at_entry_ 9 PL
get_to_cl_$unclaimed_signal 2 PL
ios_utility_$util_detach 5 PL
ios_$get_ate 66 ML
ios_utility_$util_detach 15 PL
get_to_cl_$unclaimed_signal 1 PL
ios_utility_$util_attach 4 PL
ios_$get_ate 100 ML
ios_utility_$util_attach 4 PL
ios_$get_ate 90 ML
ios_utility_$util_attach 10 PL
get_to_cl_$unclaimed_signal 1 PL
ios_utility_$util_detach 5 PL
ios_$get_ate 83 ML
ios_utility_$util_detach 15 PL
get_to_cl_$unclaimed_signal 1 PL
ios_utility_$util_attach 4 PL
ios_$get_ate 100 ML
ios_utility_$util_attach 4 PL
ios_$get_ate 90 ML
ios_utility_$util_attach 10 PL
get_to_cl_$unclaimed_signal 1 PL
ios_utility_$util_detach 5 PL
ios_$get_ate 100 ML
ios_utility_$util_detach 15 PL
get_to_cl_$unclaimed_signal 1 PL
ios_utility_$util_attach 4 PL
ios_$get_ate 100 ML
ios_utility_$util_attach 4 PL
ios_$get_ate 90 ML
ios_utility_$util_attach 20 PL
get_to_cl_$unclaimed_signal 2 PL
ios_$changemode 79 ML
rest_of_ttydim_$changemode 6 PL

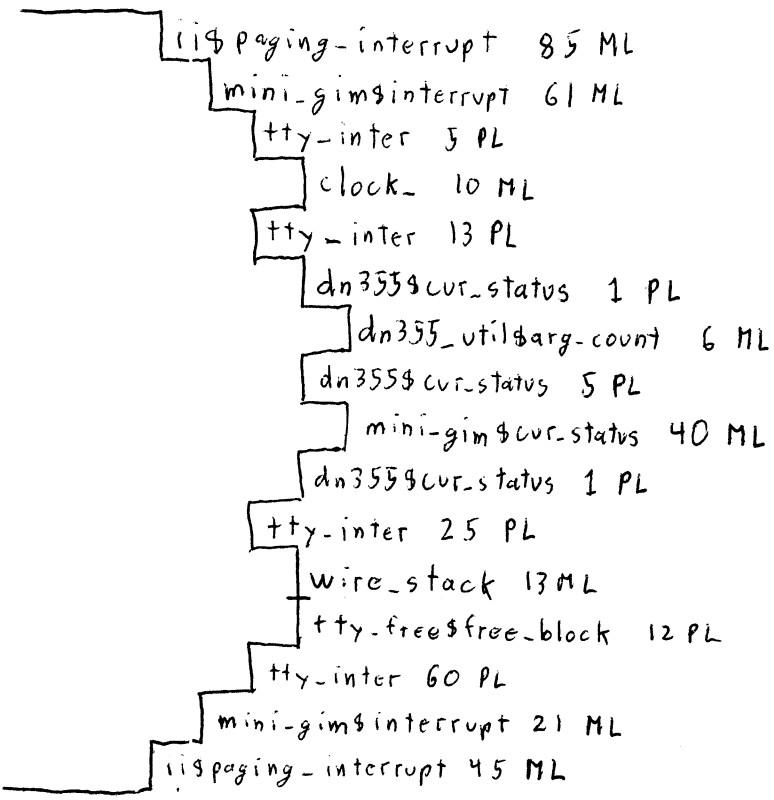

fim 123 ML
gate$gate_in 108 ML
validate_arg 205 PL
gate$gate_in 15 ML
pxss$usage_values 12 ML
gate$gate_in 6 ML
[redacted]tty-order 106 PL
cv_bin_$dec 27 PL
tty-order 15 PL
gate$gate_in 9 ML
pxss$usage_values 12 ML
gate$gate_in 37 ML
fim 33 ML
rest_of_ttydim_$changemode 1 PL
ios_$changemode 3 ML
get_to_cl_$unclaimed_signal 1 PL
listen$release_stack 8 PL
establish_cleanup_proc 35 ML
listen$release_stack 3 PL
print_ready_message_ 7 PL
clock_ 10 ML
print_ready_message_ 7 PL
cpu_time_and_paging_ 9 ML
print_ready_message 4 PL
ioa_$nnl 5 PL
tormline_ 1828 ML
ioa_$nnl 11 PL
ios_$write_ptr 57 ML
ttydim_$ttw_write 10PL
fim 123 ML
gate$gate_in 109 ML
validate_arg 179 PL
gate$gate_in 15 ML
pxss$usage_values 12 ML
gate$gate_in 6 ML
tty-write 21 PL
tty-inter$prescan 7 PL
dn355$cur_status 1 PL
dn355_util$arg_count 6 ML
dn355$cur_status 5 PL
mini_gim$cur_status 40 ML


dn355$cur_status 1 PL
tty-inter$prescan 25 PL
tty-write 748 PL
tty-inter$write 6 PL
dn355$cur_status 1 PL
dn355_util$arg_count 6 ML
dn355$cur_status 5 PL
mini_gim$cur_status 40 ML
dn355$cur_status 1 PL
tty-inter$write 16 PL
tty-write 4 PL
gate$gate_in 9 ML
pxss$usage_values 12 ML
gate$gate_in 37 ML
fim 33 ML
ttydim-$tw_write 6 PL
ios-$write_ptr 3 ML
ioa-$nnl 1 PL
print_ready_message_ 1 PL
listen_$release_stack 1 PL
fim 123 ML
gate$gate_in 108 ML
pxss$usage_values 12 ML
gate$gate_in 5 ML
page$reset_working_set 15 ML
page_util$reset_working_set 30 ML
page$reset_working_set 2 ML
gate$gate_in
pxss$usage_values 12 ML
gate$gate_in 37 ML
fim 33 ML
listen_$release_stack 3 PL
ios_$read_ptr 43 ML
ttydim_$tw_read 8 PL
fim 123 ML
gate$gate_in 108 ML
validate_arg 97 PL
gate$gate_in 12 ML
pxss$usage_values 12 ML
gate$gate_in 5 ML
tty_read 34 PL

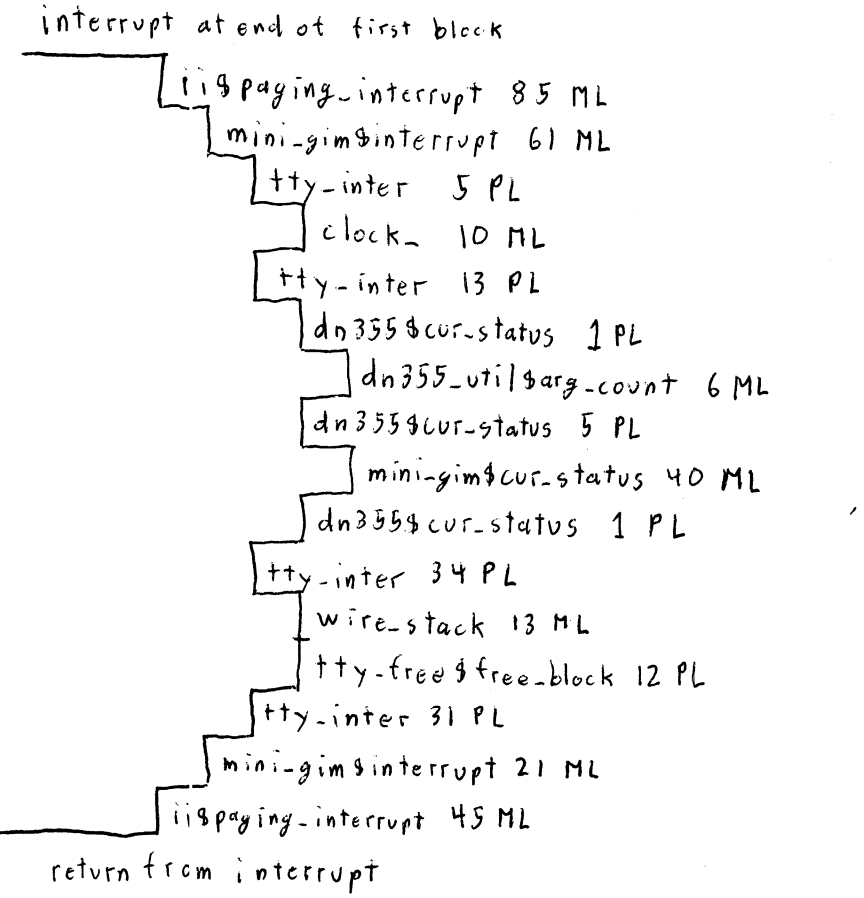


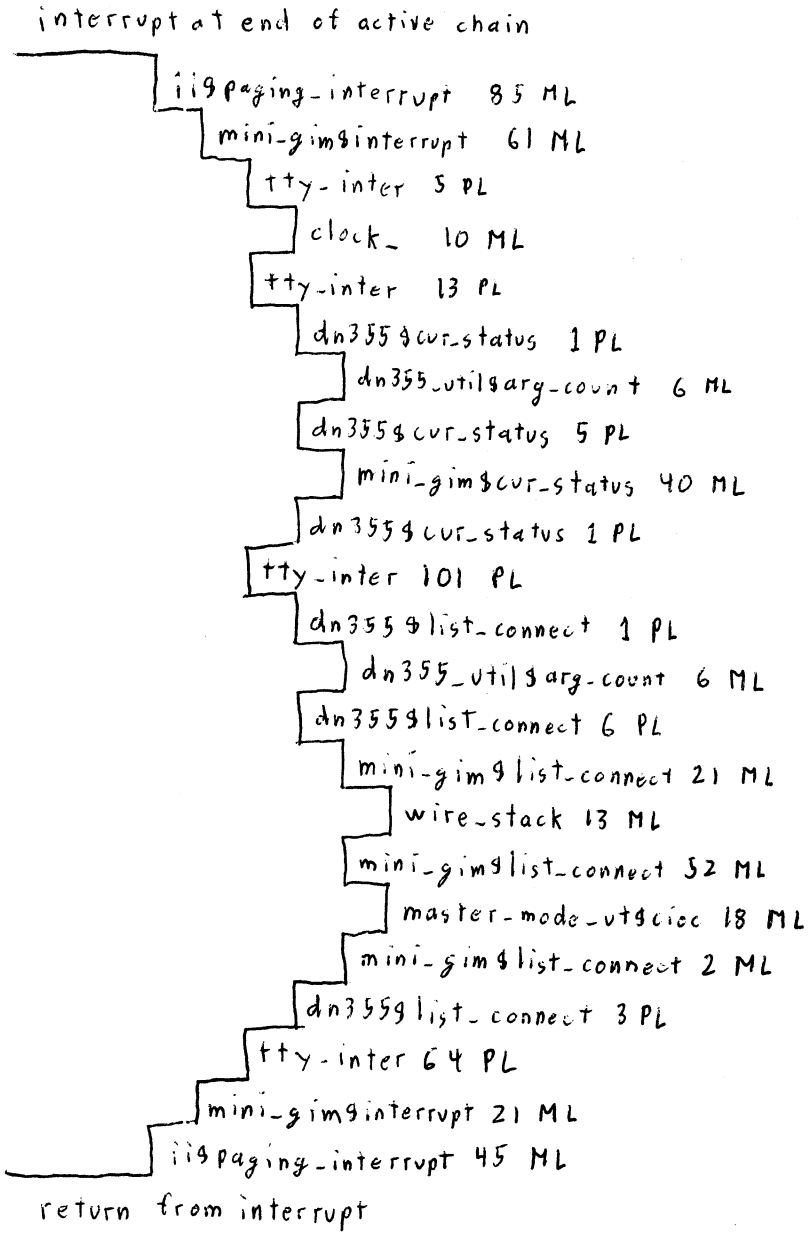
give up the processor

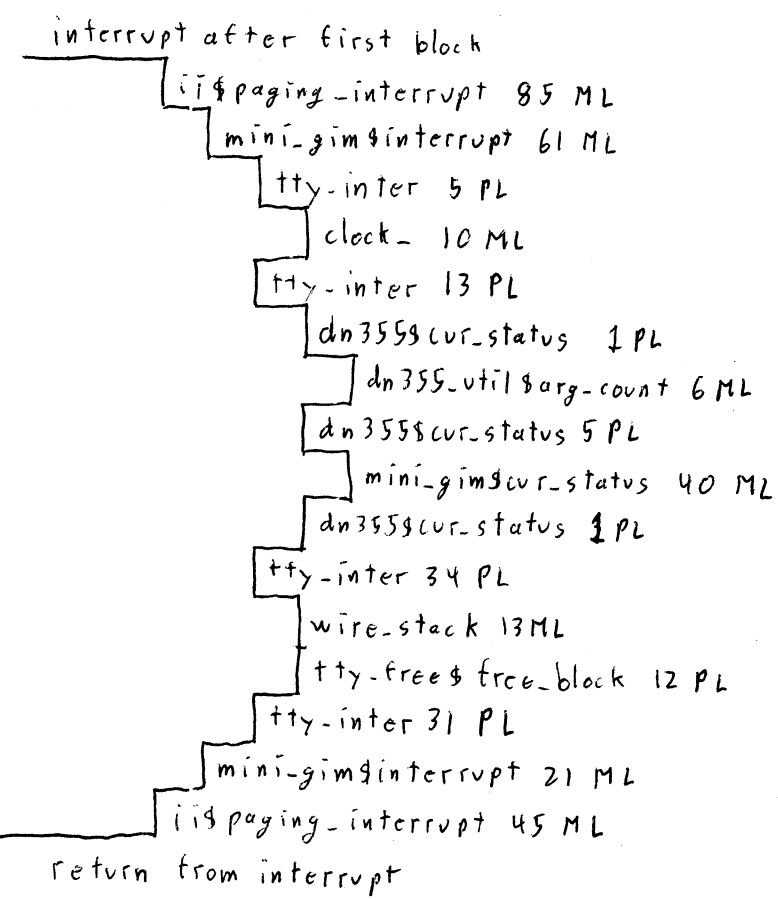
interrupt after addressing printer

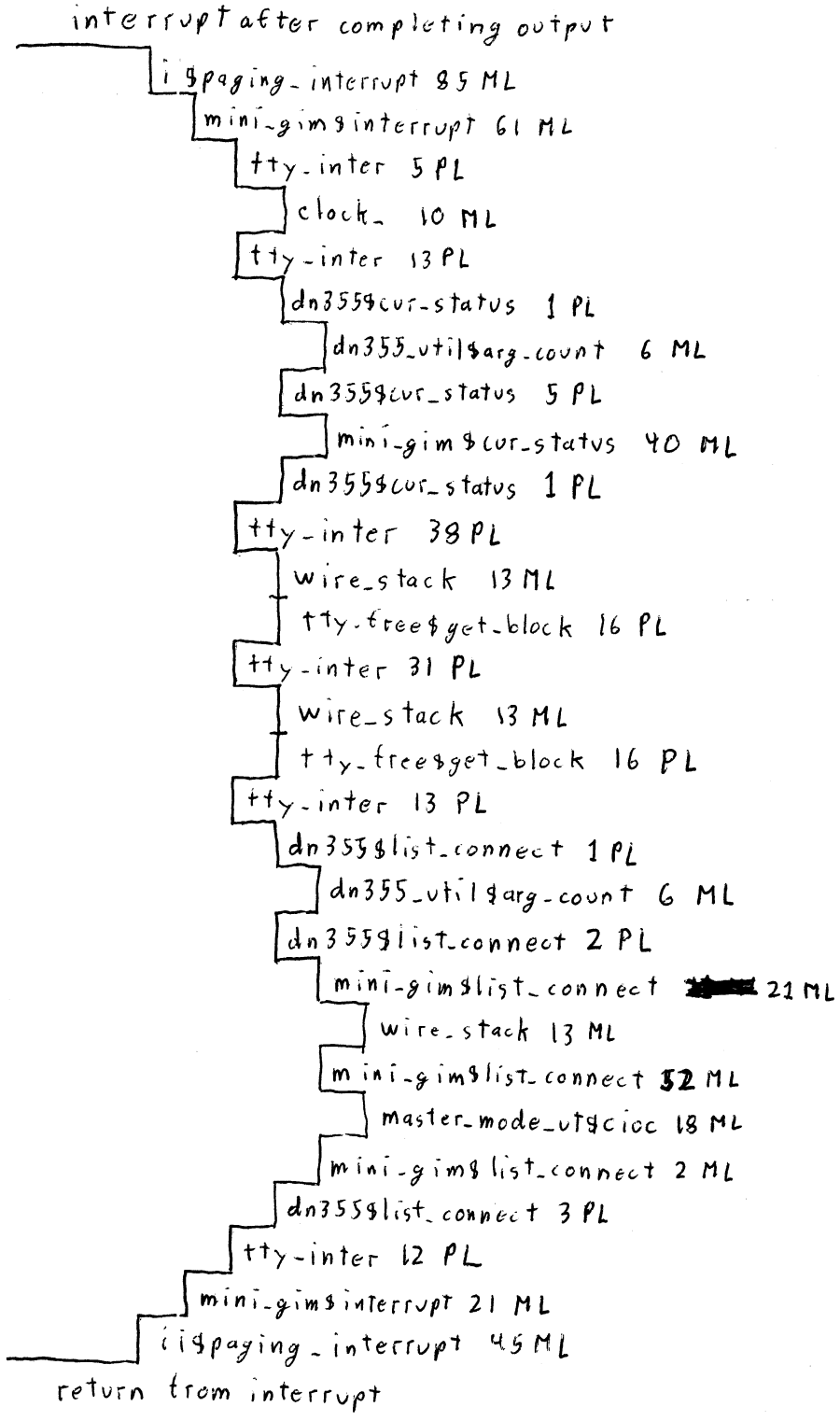


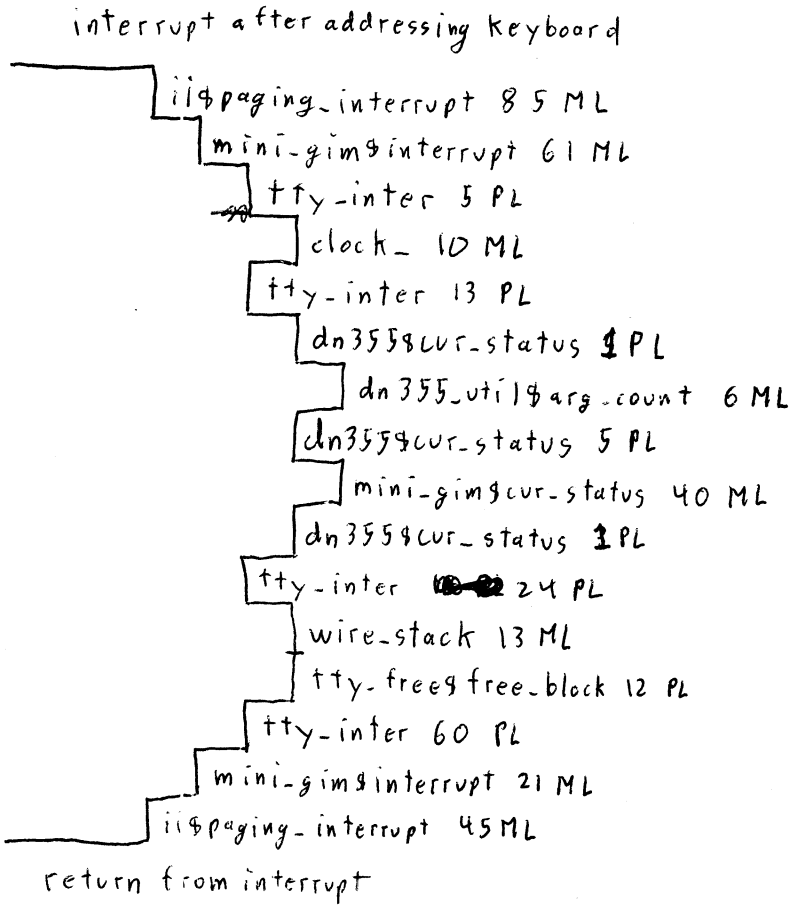
return from interrupt

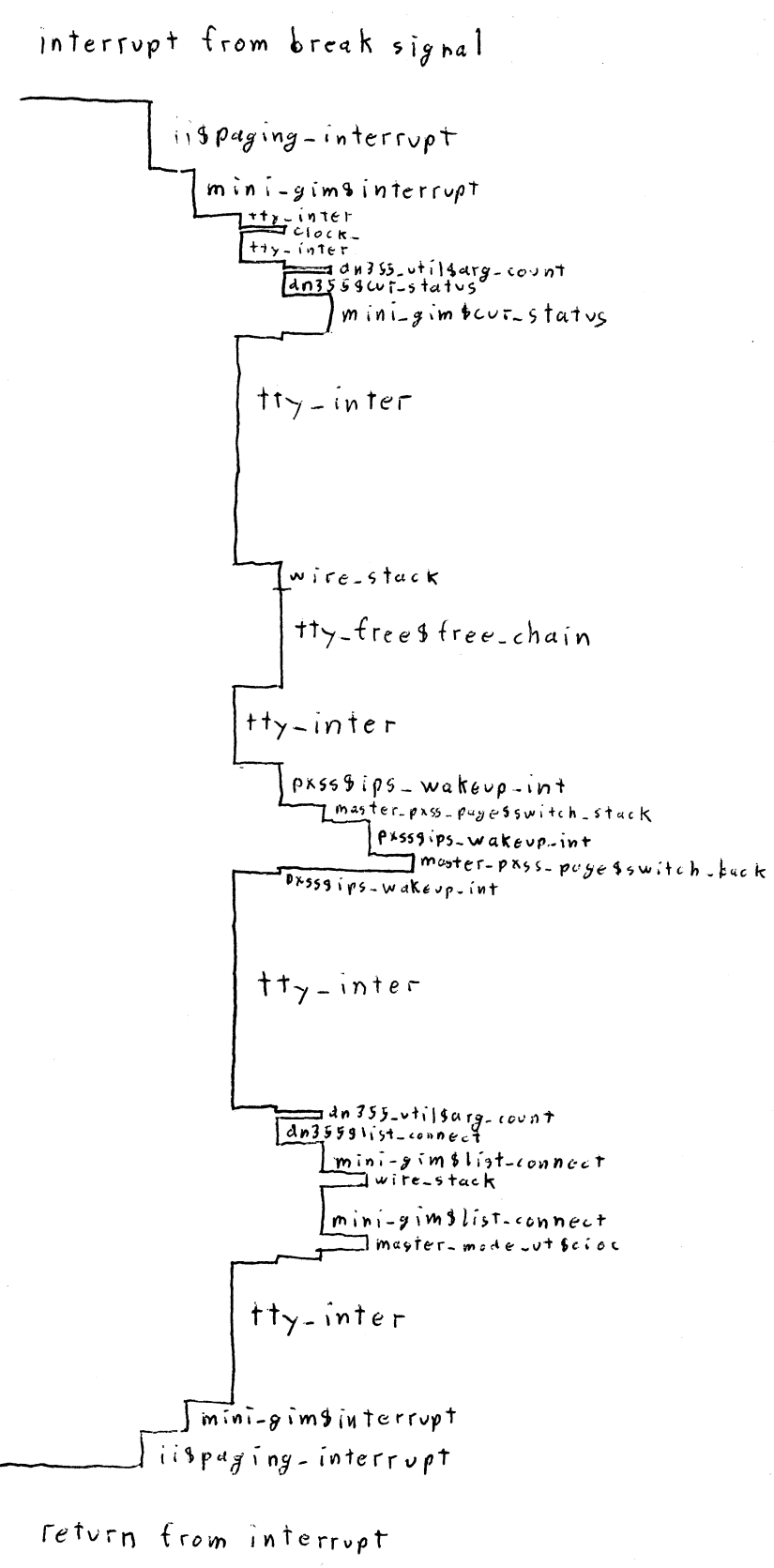

interrupt at end of first block
ii$paging_interrupt 85 ML
mini_gim$interrupt 61 ML
tty_inter 5 PL
clock_ 10 ML
tty_inter 13 PL
dn355$cur_status 1 PL
dn355_util$arg_count 6 ML
dn355$cur_status 5 PL
mini_gim$cur_status 40 ML
dn355$cur_status 1 PL
tty_inter 34 PL
wire_stack 13 ML
tty_free$free_block 12 PL
tty_inter 31 PL
mini_gim$interrupt 21 ML
ii$paging_interrupt 45 ML
return from interrupt

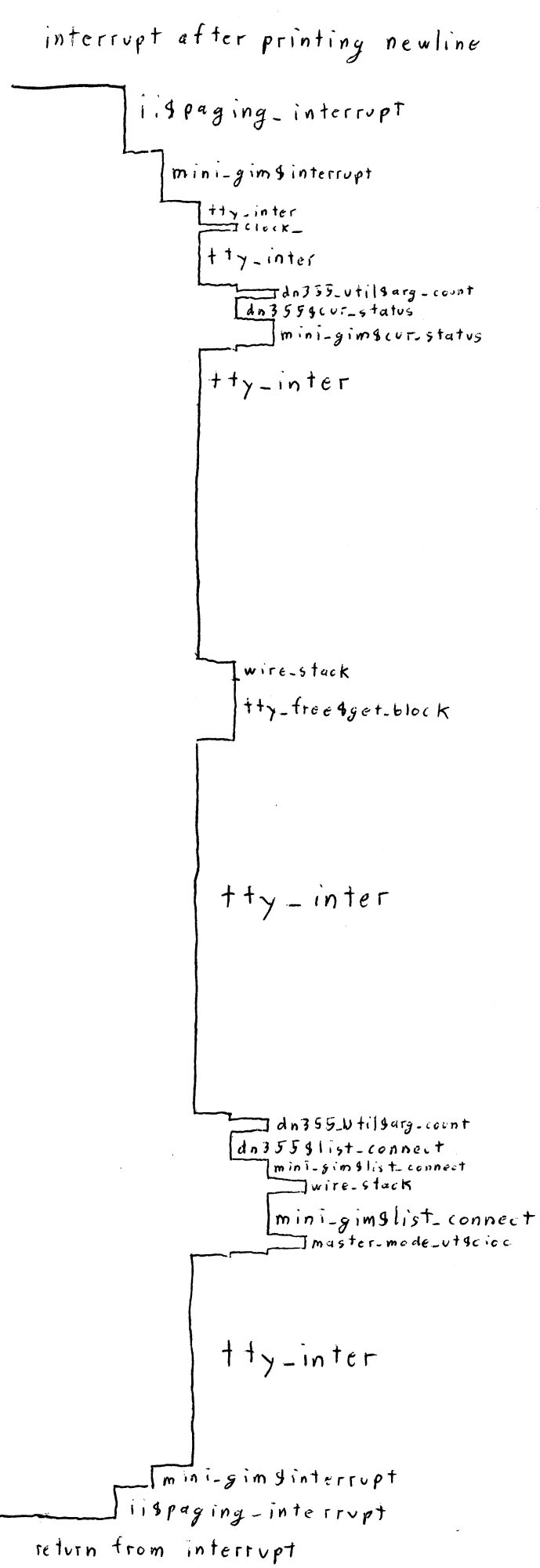

interrupt at end of active chain
iig$paging_interrupt 85 ML
mini_gim$interrupt 61 ML
tty_inter 5 PL
clock_ 10 ML
tty_inter 13 PL
dn355$cur_status 1 PL
dn355_util$arg_count 6 ML
dn355$cur_status 5 PL
mini_gim$cur_status 40 ML
dn355$cur_status 1 PL
tty_inter 101 PL
dn355$list_connect 1 PL
dn355_util$arg_count 6 ML
dn355$list_connect 6 PL
mini_gim$list_connect 21 ML
wire_stack 13 ML
mini_gim$list_connect 52 ML
master_mode_vt$ciec 18 ML
mini_gim$list_connect 2 ML
dn355$list_connect 3 PL
tty_inter 64 PL
mini_gim$interrupt 21 ML
iig$paging_interrupt 45 ML
return from interrupt


interrupt after first block
ii$paging_interrupt 85 ML
mini_gim$interrupt 61 ML
tty_inter 5 PL
clock_ 10 ML
tty_inter 13 PL
dn355$cur_status 1 PL
dn355_util$arg_count 6 ML
dn355$cur_status 5 PL
mini_gim$cur_status 40 ML
dn355$cur_status 1 PL
tty_inter 34 PL
wire_stack 13 ML
tty_free$free_block 12 PL
tty_inter 31 PL
mini_gim$interrupt 21 ML
ii$paging_interrupt 45 ML
return from interrupt

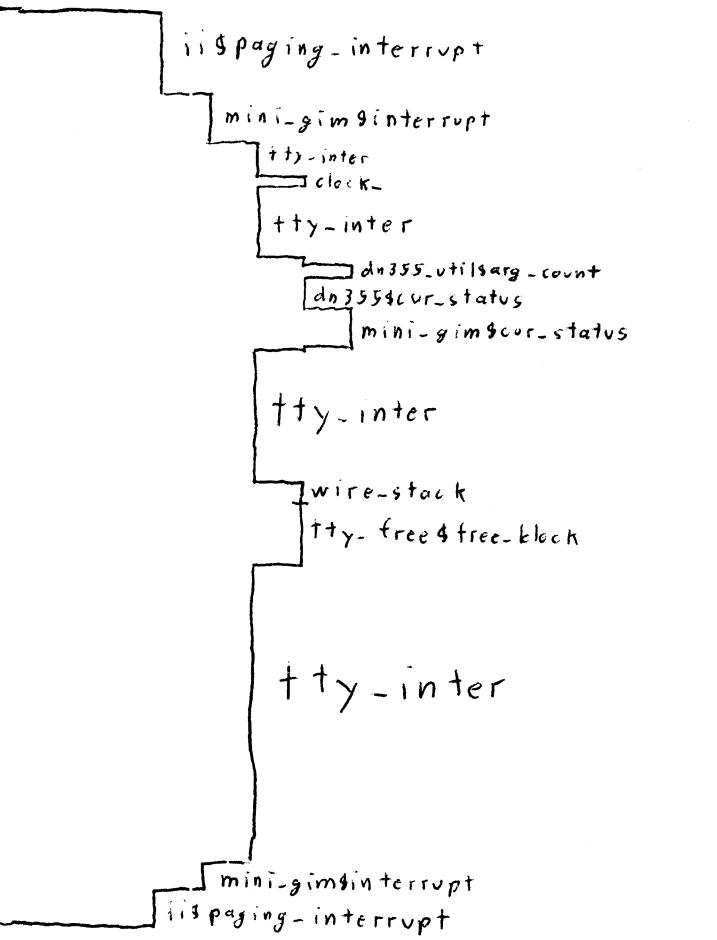

interrupt after completing output
i$paging_interrupt 85 ML
mini_gim$interrupt 61 ML
tty_inter 5 PL
clock_ 10 ML
tty_inter 13 PL
dn355$cur_status 1 PL
dn355_util$arg_count 6 ML
dn355$cur_status 5 PL
mini_gim$cur_status 40 ML
dn355$cur_status 1 PL
tty_inter 38 PL
wire_stack 13 ML
tty_free$get_block 16 PL
tty_inter 31 PL
wire_stack 13 ML
tty_free$get_block 16 PL
tty_inter 13 PL
dn355$list_connect 1 PL
dn355_util$arg_count 6 ML
dn355$list_connect 2 PL
mini_gim$list_connect [redacted] 21 ML
wire_stack 13 ML
mini_gim$list_connect 52 ML
master_mode_ut$cioc 18 ML
mini_gim$list_connect 2 ML
dn355$list_connect 3 PL
tty_inter 12 PL
mini_gim$interrupt 21 ML
i$paging_interrupt 45 ML
return from interrupt


interrupt after addressing keyboard
ii$paging_interrupt 85 ML
mini_gim$interrupt 61 ML
tty_inter 5 PL
clock_ 10 ML
tty_inter 13 PL
dn355$cur_status 1 PL
dn355_util$arg_count 6 ML
dn355$cur_status 5 PL
mini_gim$cur_status 40 ML
dn355$cur_status 1 PL
tty_inter 24 PL
wire_stack 13 ML
tty_free$free_block 12 PL
tty_inter 60 PL
mini_gim$interrupt 21 ML
ii$paging_interrupt 45 ML
return from interrupt

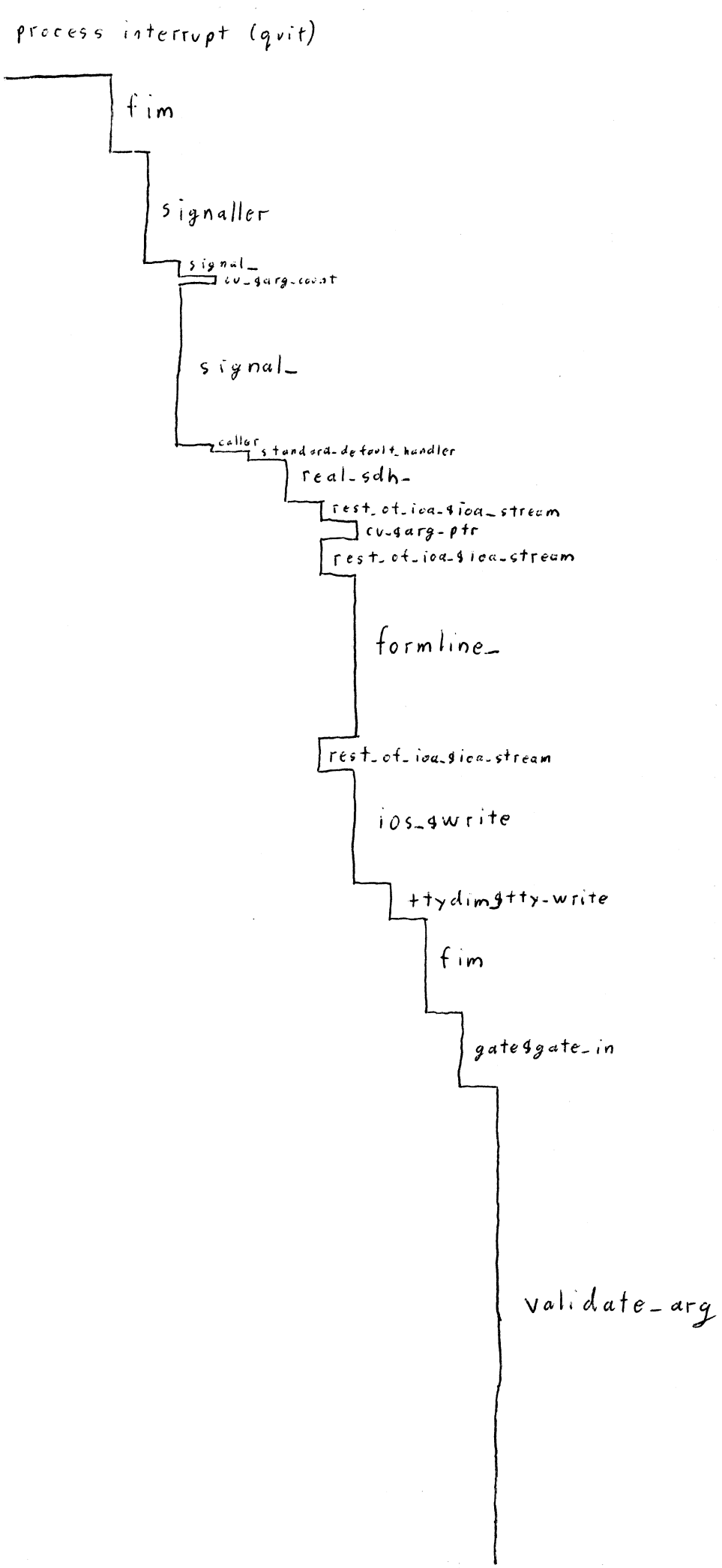

interrupt from break signal
ii$paging_interrupt
mini_gim$interrupt
tty_inter
clock_
tty_inter
dn355_util$arg_count
dn355$cut_status
mini_gim$cut_status
tty_inter
wire_stack
tty_free$free_chain
tty_inter
pxss$ips_wakeup_int
master_pxss_page$switch_stack
pxss$ips_wakeup_int
master_pxss_page$switch_back
pxss$ips_wakeup_int
tty_inter
dn355_util$arg_count
dn355$list_connect
mini_gim$list_connect
wire_stack
mini_gim$list_connect
master_mode_ut$ioc
tty_inter
mini_gim$interrupt
ii$paging_interrupt
return from interrupt

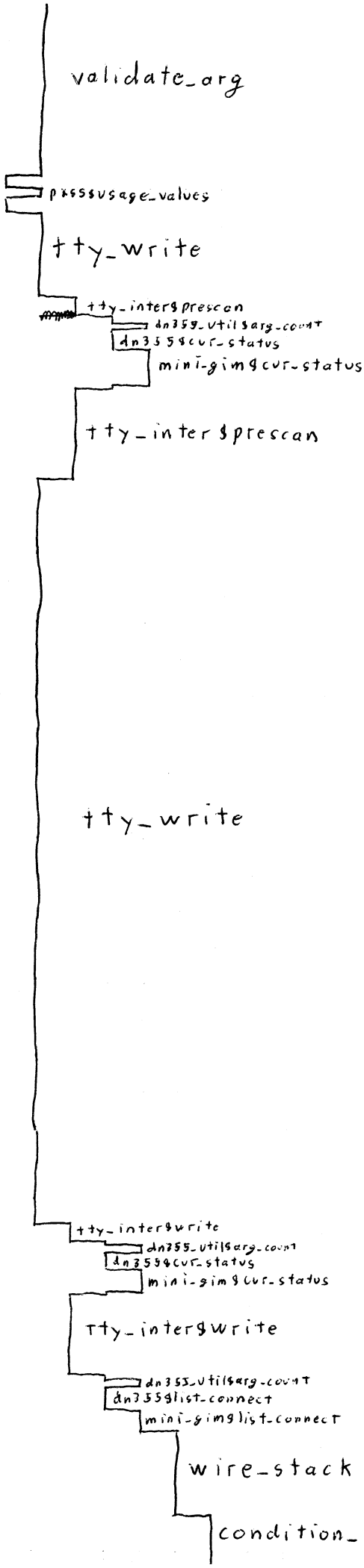

interrupt after printing newline
i.$paging_interrupt
mini_gim$interrupt
tty_inter
clock_
tty_inter
dn355_util$arg_count
dn355$cur_status
mini_gim$cur_status
tty_inter
wire_stack
tty_free$get_block
tty_inter
dn355_util$arg_count
dn355$list_connect
mini_gim$list_connect
wire_stack
mini_gim$list_connect
master_mode_ut$cicc
tty_inter
mini_gim$interrupt
i.$paging_interrupt
return from interrupt

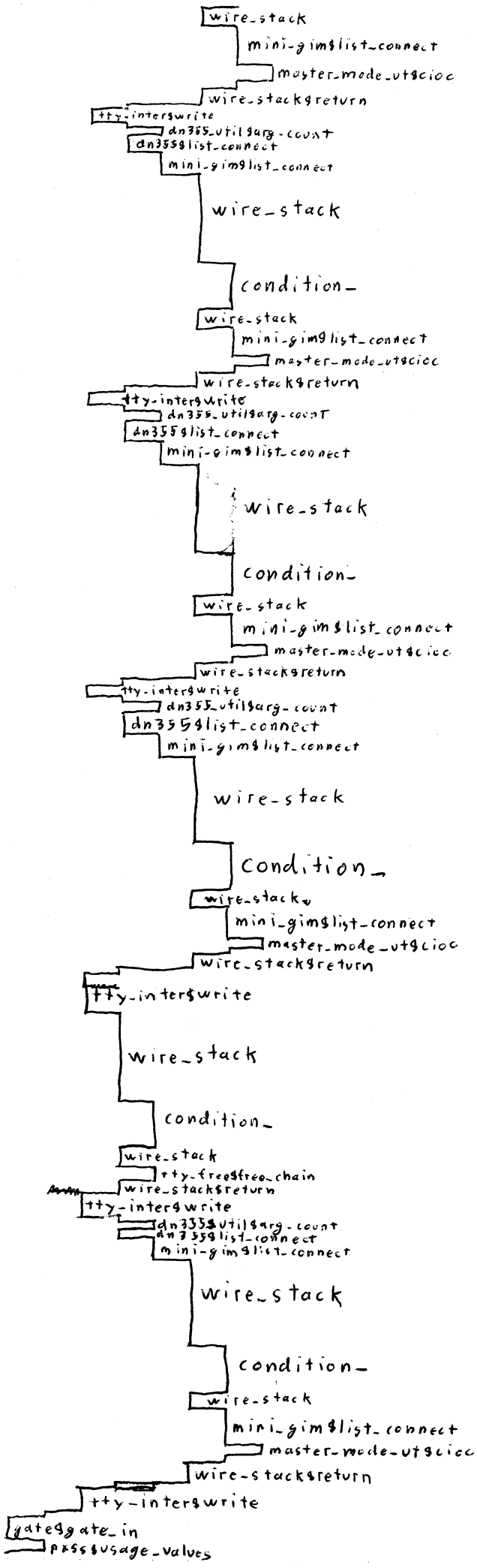
interrupt after addressing keyboard

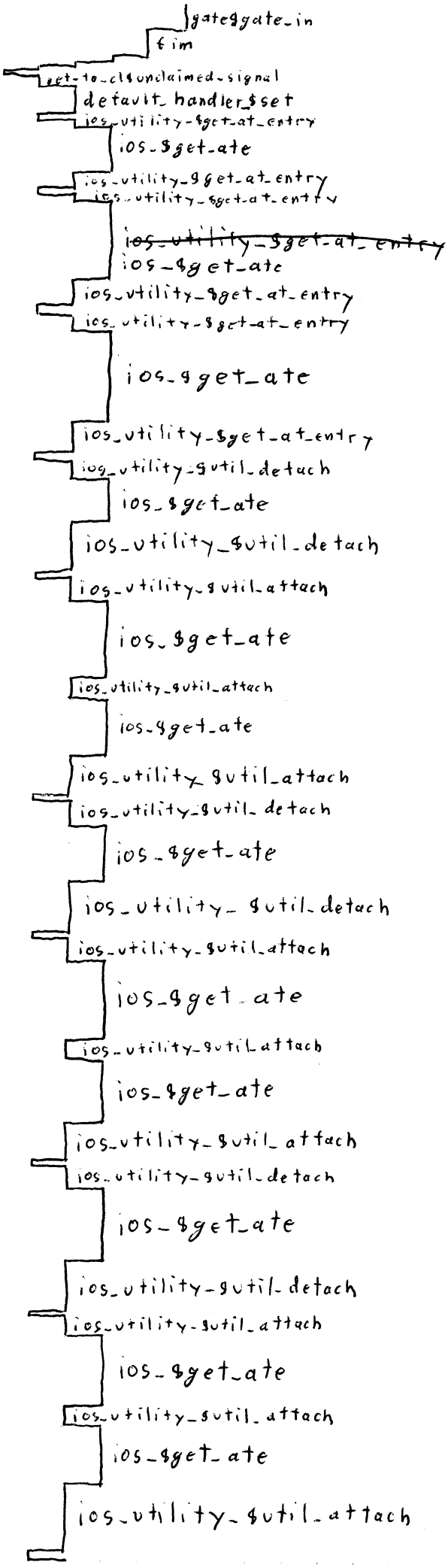


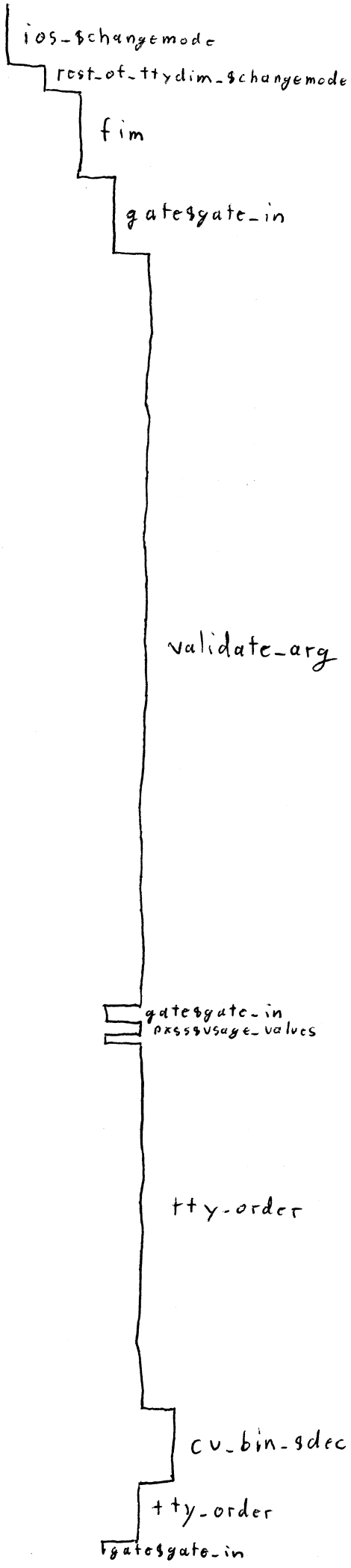
return to interrupted program

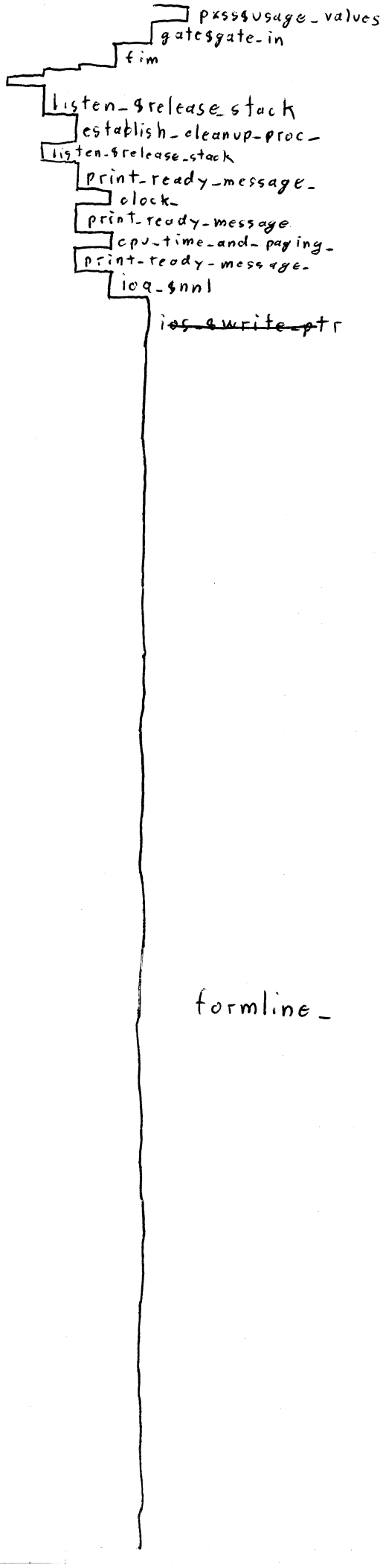

process interrupt (qvit)
fim
signaller
signal_
cv_garg_const
signal_
caller
standard_default_handler
real_sdh_
rest_of_ioa_$ioa_stream
cv_garg_ptr
rest_of_ioa_$ioa_stream
formline_
rest_of_ioa_$ioa_stream
ios_$write
ttydim$tty_write
fim
gate$gate_in
validate_arg

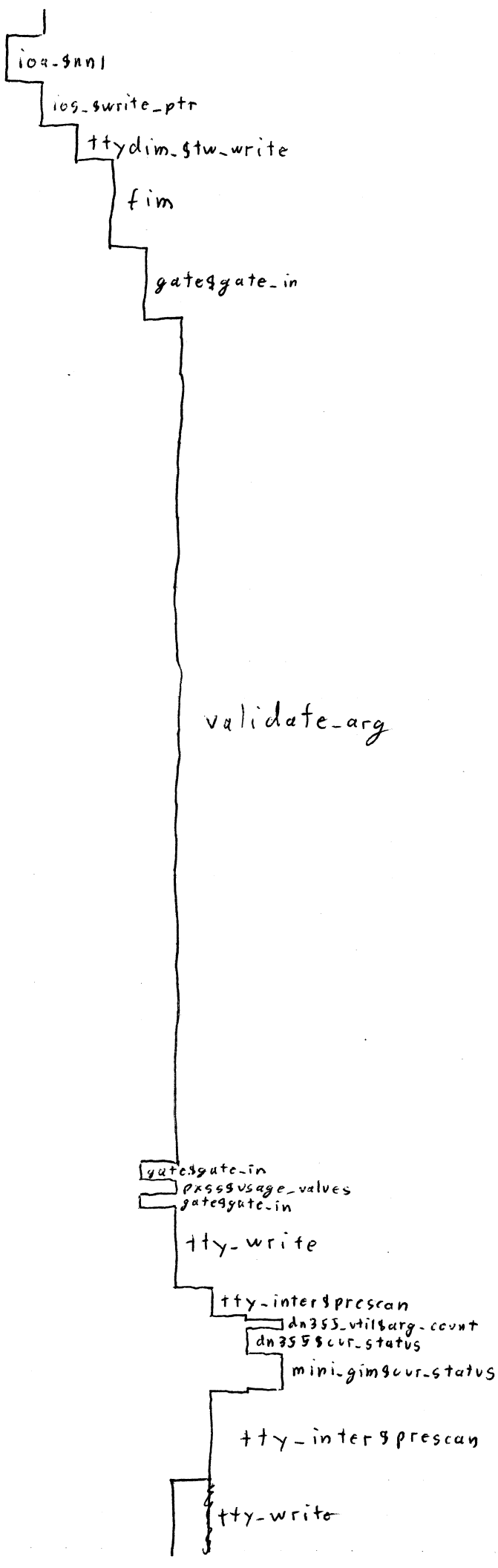

validate_arg
print_usage_values
tty_write
tty_intern$prescan
dn355_util$arg_count
dn355$cur_status
mini_gim$cur_status
tty_intern$prescan
tty_write
tty_intern$write
dn355_util$arg_count
dn355$cur_status
mini_gim$cur_status
tty_intern$write
dn355_util$arg_count
dn355$list_connect
mini_gim$list_connect
wire_stack
condition_


wire_stack
mini_gim$list_connect
master_mode_ut$cioc
wire_stack$return
tty_inter$write
dn355_util$arg_count
dn355$list_connect
mini_gim$list_connect
wire_stack
condition_
wire_stack
mini_gim$list_connect
master_mode_ut$cioc
wire_stack$return
tty_inter$write
dn355_util$arg_count
dn355$list_connect
mini_gim$list_connect
wire_stack
condition_
wire_stack
mini_gim$list_connect
master_mode_ut$cioc
wire_stack$return
tty_inter$write
dn355_util$arg_count
dn355$list_connect
mini_gim$list_connect
wire_stack
condition_
wire_stack
mini_gim$list_connect
master_mode_ut$cioc
wire_stack$return
tty_inter$write
wire_stack
condition_
wire_stack
tty_free$free_chain
wire_stack$return
tty_inter$write
dn335$util$arg_count
dn355$list_connect
mini_gim$list_connect
wire_stack
condition_
wire_stack
mini_gim$list_connect
master_mode_ut$cioc
wire_stack$return
tty_inter$write
gate$gate_in
pass$usage_values

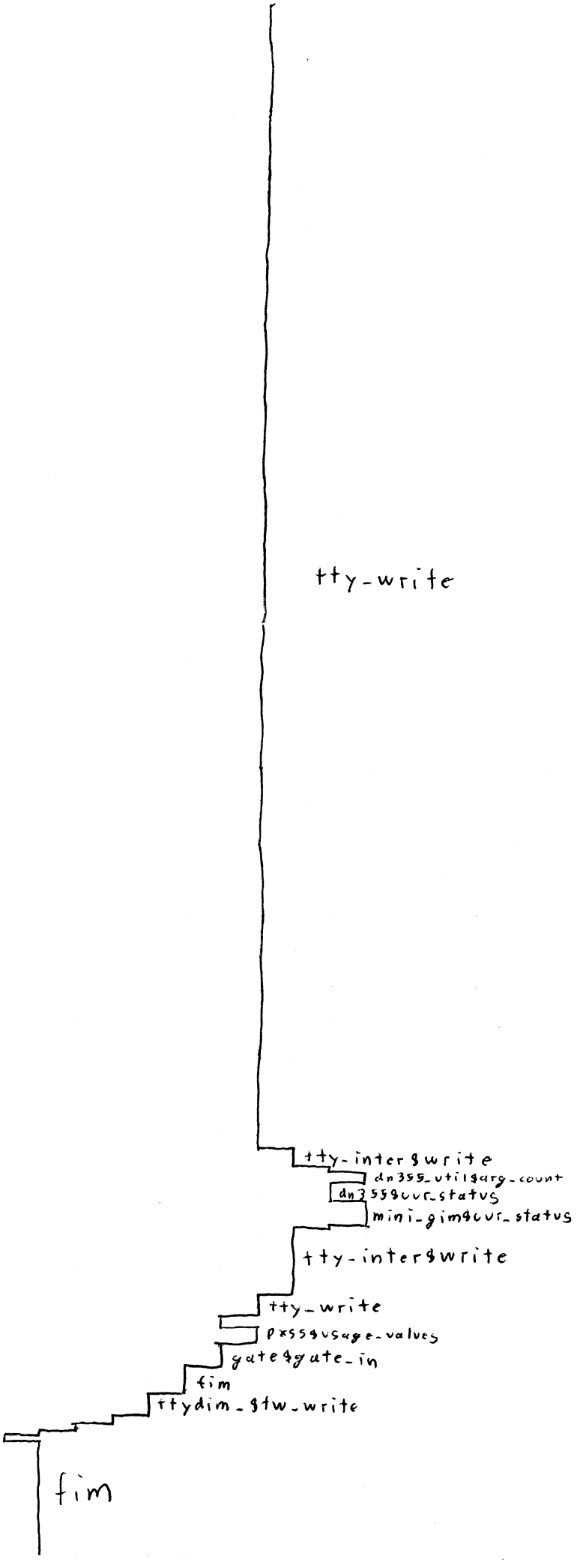

gate$gate_in
tim
set_to_clsunclaimed_signal
default_handler_$set
ios_utility_$get_at_entry
ios_$get_ate
ios_utility_$get_at_entry
ios_utility_$get_at_entry
ios_utility_$get_at_entry
ios_$get_ate
ios_utility_$get_at_entry
ios_utility_$get_at_entry
ios_$get_ate
ios_utility_$get_at_entry
ios_utility_$util_detach
ios_$get_ate
ios_utility_$util_detach
ios_utility_$util_attach
ios_$get_ate
ios_utility_$util_attach
ios_$get_ate
ios_utility_$util_attach
ios_utility_$util_detach
ios_$get_ate
ios_utility_$util_detach
ios_utility_$util_attach
ios_$get_ate
ios_utility_$util_attach
ios_$get_ate
ios_utility_$util_attach
ios_utility_$util_detach
ios_$get_ate
ios_utility_$util_detach
ios_utility_$util_attach
ios_$get_ate
ios_utility_$util_attach
ios_$get_ate
ios_utility_$util_attach

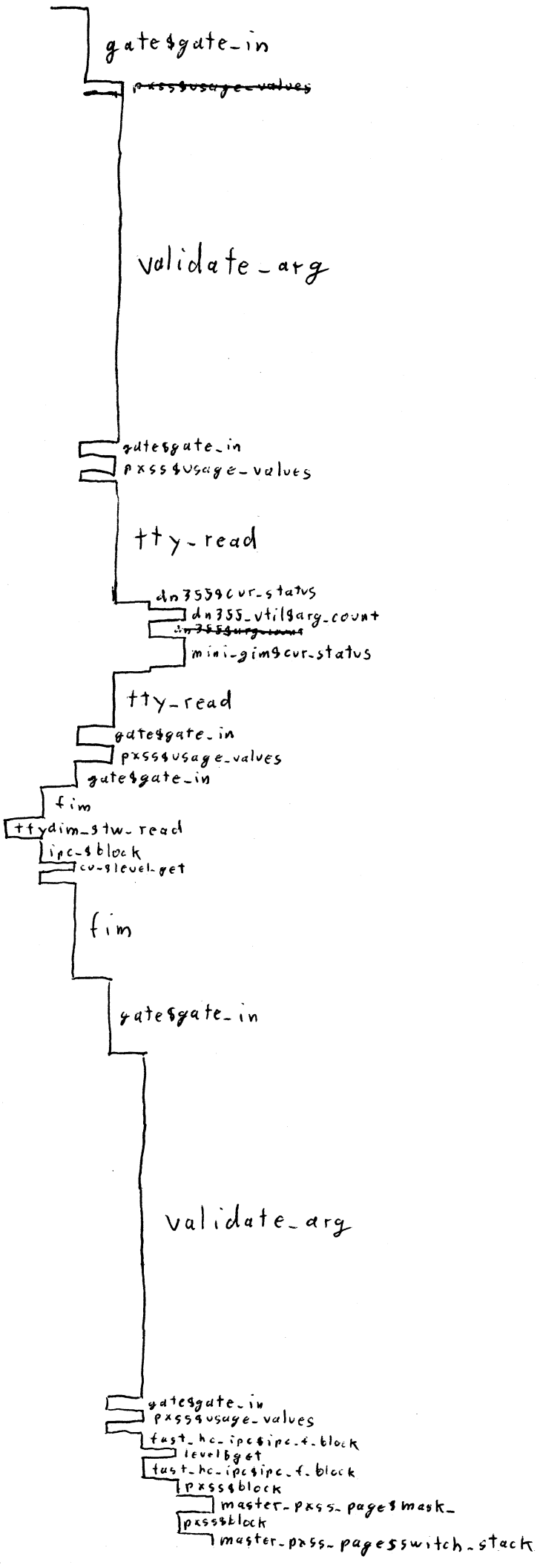

ios_$changemode
rest_of_ttydim_$changemode
fim
gate$gate_in
validate_arg
gate$gate_in
proc$usage_values
tty_order
cv_bin_$dec
tty_order
gate$gate_in

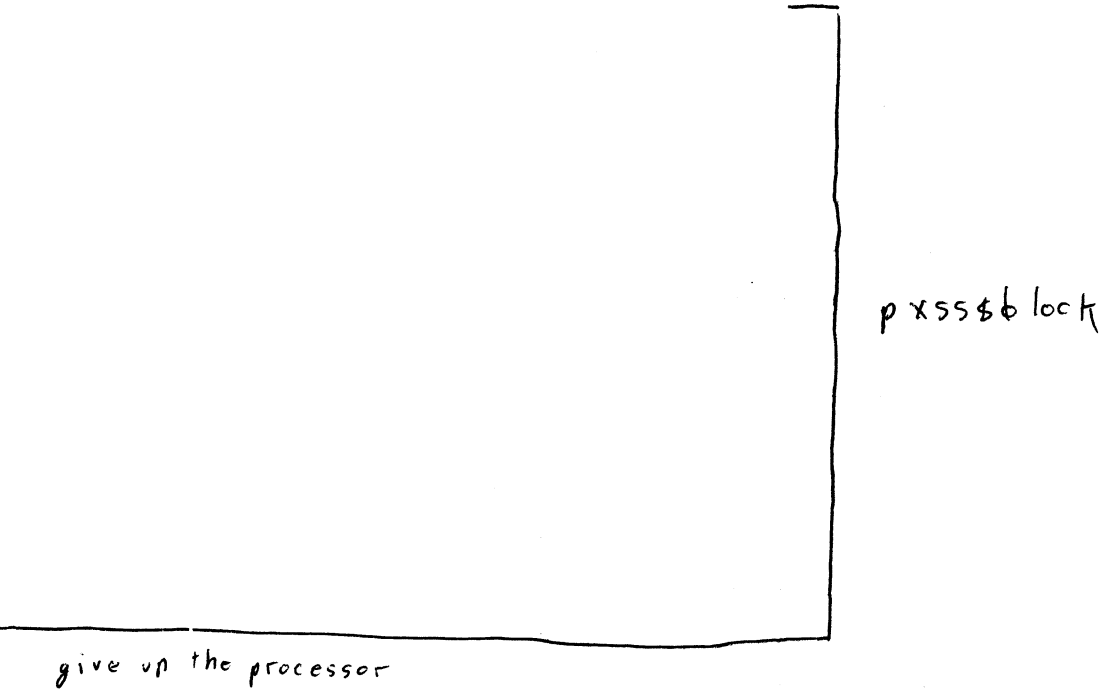

pxss$usage_values
gatesgate_in
tim
listen_$release_stack
establish_cleanup_proc_
listen_$release_stack
print_ready_message_
clock_
print_ready_message
cpu_time_and_paging_
print_ready_message_
ioa_$nnl
~~ios_$write_ptr~~
formline_

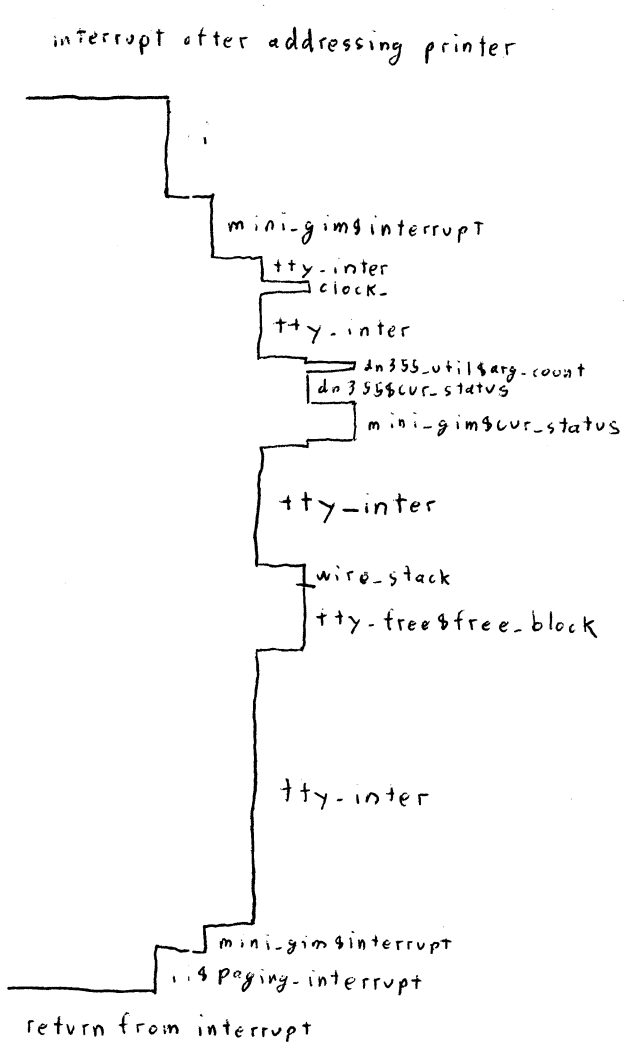

ioa_$nnl
ios_$write_ptr
ttydim_$tw_write
fim
gate$gate_in
validate_arg
gate$gate_in
pxss$vsage_values
gate$gate_in
tty_write
tty_inter$prescan
dn355_util$arg_count
dn355$cur_status
mini_gim$cur_status
tty_inter$prescan
tty_write

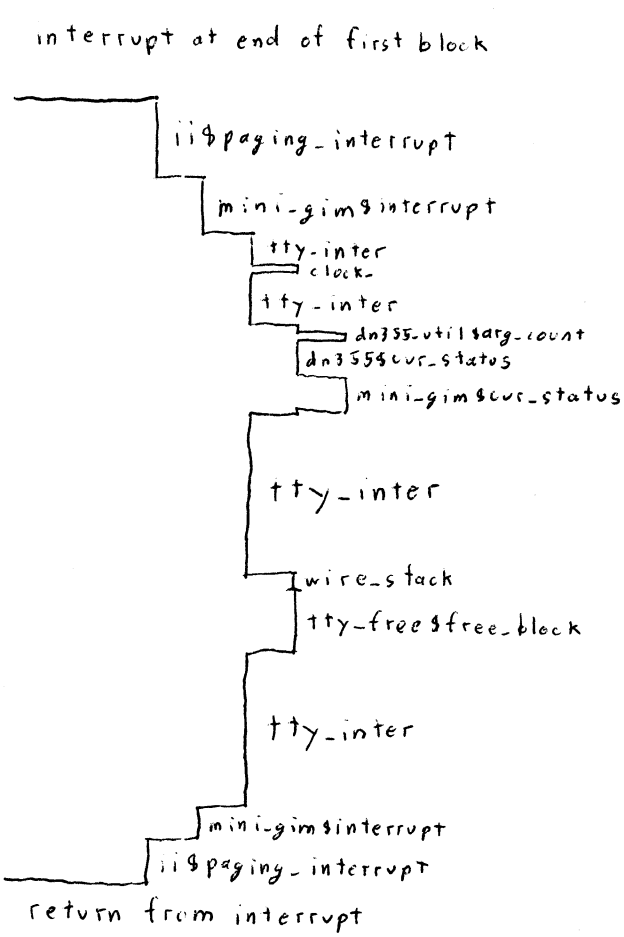
tty_write

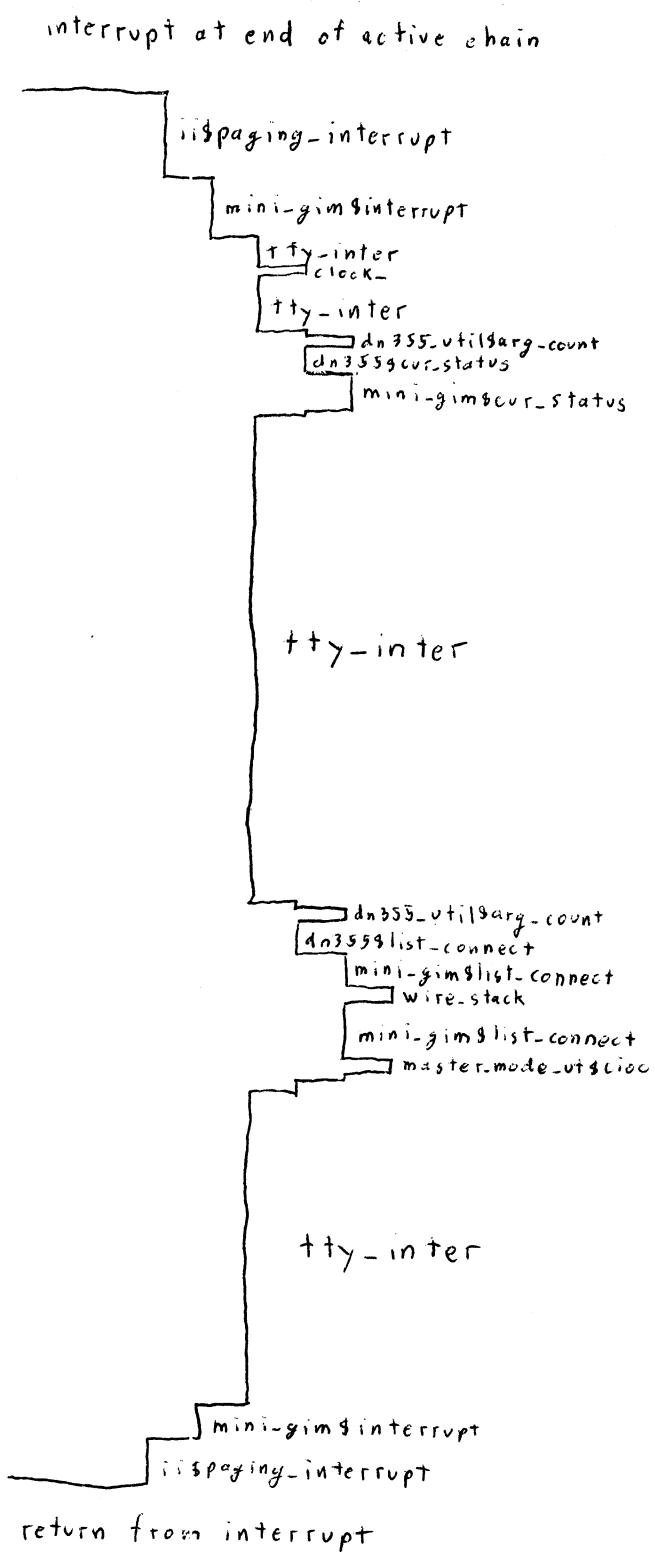

tty_write
tty_inter$write
dn355_util$arg_count
dn355$cur_status
mini_gim$cur_status
tty_inter$write
tty_write
pxss$usage_values
gate$gate_in
fim
ttydim_$tw_write
fim


gate$gate_in
pxss$usage_values
validate_arg
gate$gate_in
pxss$usage_values
tty_read
dn355$cur_status
dn355_util$arg_count
mini_gim$cur_status
tty_read
gate$gate_in
pxss$usage_values
gate$gate_in
fim
ttydim_stw_read
ipc_$block
cv_$level_get
fim
gate$gate_in
validate_arg
gate$gate_in
pxss$usage_values
fast_hc_ipc$ipc_f_block
level$get
fast_hc_ipc$ipc_f_block
pxss$block
master_pxss_page$mask_
pxss$block
master_pxss_page$switch_stack

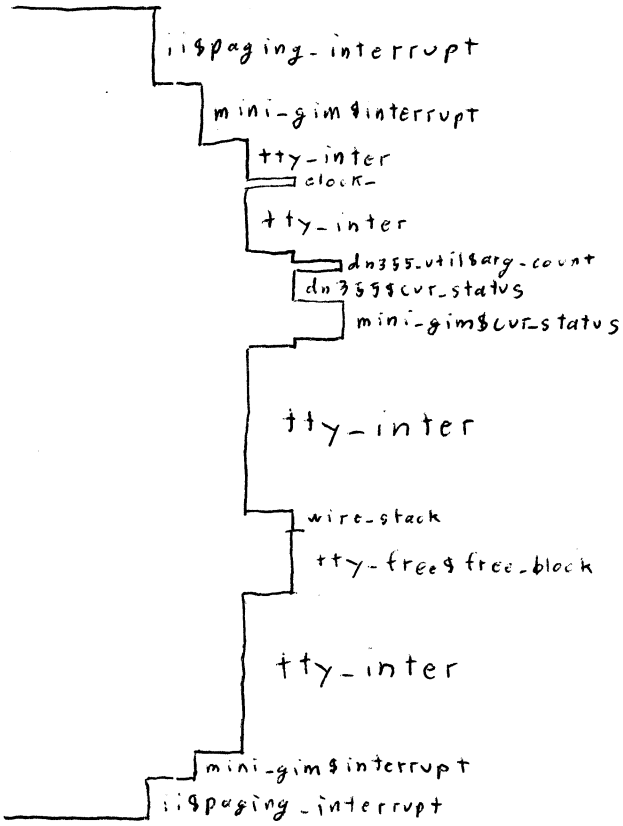

pxssblock
give up the processor


interrupt after addressing printer
mini_gim$interrupt
tty_inter
clock_
tty_inter
dn355_util$arg_count
dn355$cur_status
mini_gim$cur_status
tty_inter
wire_stack
tty_free$free_block
tty_inter
mini_gim$interrupt
...$paging_interrupt
return from interrupt

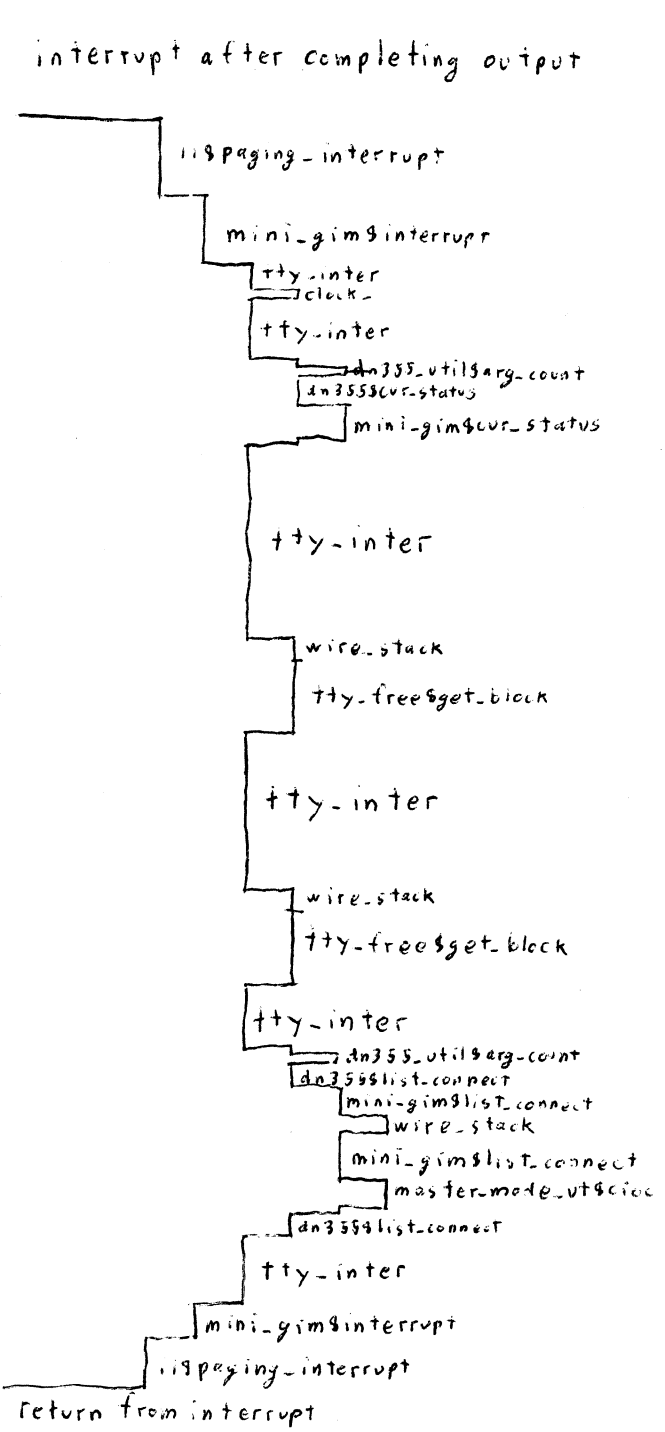

interrupt at end of first block
ii$paging_interrupt
mini_gim$interrupt
tty_inter
clock_
tty_inter
dn355_util$arg_count
dn355$cur_status
mini_gim$cur_status
tty_inter
wire_stack
tty_free$free_block
tty_inter
mini_gim$interrupt
ii$paging_interrupt
return from interrupt


interrupt at end of active chain
ii$paging_interrupt
mini-gim$interrupt
tty-inter
clock_
tty-inter
dn355_util$arg_count
dn355$cur_status
mini-gim$cur_status
tty-inter
dn355_util$arg_count
dn355$list_connect
mini-gim$list_connect
wire_stack
mini-gim$list_connect
master_mode_ut$cioc
tty-inter
mini-gim$interrupt
ii$paging_interrupt
return from interrupt

interrupt after first block



return from interrupt


interrupt after completing output
ii$paging_interrupt
mini_gim$interrupt
tty_inter
clock_
tty_inter
dn355_util$arg_count
dn355$cur_status
mini_gim$cur_status
tty_inter
wire_stack
tty_free$get_block
tty_inter
wire_stack
tty_free$get_block
tty_inter
dn355_util$arg_count
dn355$list_connect
mini_gim$list_connect
wire_stack
mini_gim$list_connect
master_mode_ut$ioc
dn355$list_connect
tty_inter
mini_gim$interrupt
ii$paging_interrupt
return from interrupt

interrupt after addressing keyboard