Appendix D

GDB Cheatsheet for μC++

The following gdb and Python macros allow gdb to understand the μC++ runtime environment: tasks (see Section 2.13, p. 29), processors (see Section 8.4, p. 125), and clusters (see Section 8.3, p. 123).

These instructions assume the install <prefix> for μC++ is /usr/local/u++-7.0.0; if installed elsewhere, change <prefix>. Copy <prefix>/.gdbinit to your home directory or merge it into your existing .gdbinit file. If installed elsewhere, edit the <prefix> within the .gdbinit file to the install location. Thereafter, gdb automatically loads the .gdbinit file from the home directory at start up making the following new gdb commands available.

Debugging involves setting one or more breakpoints in a program. When a breakpoint is encountered, the entire concurrent program stops, i.e., all user and kernel threads. At this point, it is possible to examine the stacks of each μC++ task stack by listing all the tasks (command task), switching to an individual task (command task 2), and printing the back trace (command backtrace) for its stack. The brace trace shows where that task is executing, and by moving up and down the back trace, it is possible to examine the variables in each stack frame.

D.1 Clusters

Command	*Description*	*Example*
clusters	print a list of all clusters	clusters

D.2 Processors

Command	*Description*	*Example*
processors	print all info about all processors on all clusters	processors
processors *clusterName*	print all info about all processors on a particular cluster	processors userCluster

D.3 Tasks

Command	*Description*	*Example*
task	print application tasks in userCluster	task
task *clusterName*	print application tasks in a clusterName	task clusterA
task all	print all tasks in all clusters	task all
task *id*	context switch to task *id* on userCluster	task 2
task 0x*address*	context switch to a task based on its address on any cluster	task 0x80024875
task *id clusterName*	context switch to a task on a specified cluster	task 2 clusterA
prevtask	switch back to the previous task on the stack	prevtask