

## Arvados - Bug #17413

### [arv-mount] Exit (instead of undefined behavior) after unhandled exception

02/18/2021 07:43 PM - Tom Clegg

<b>Status:</b>	New	<b>Start date:</b>	
<b>Priority:</b>	Normal	<b>Due date:</b>	
<b>Assigned To:</b>		<b>% Done:</b>	0%
<b>Category:</b>	FUSE	<b>Estimated time:</b>	0.00 hour
<b>Target version:</b>			
<b>Description</b>			
When arv-mount encounters an unhandled exception during a fuse operation, it returns a generic I/O error.			
<pre>except:     _logger.exception("Unhandled exception during FUSE operation")     raise llfuse.FUSEError(errno.EIO)</pre>			
Since arv-mount uses "finally", function decorators, etc., this should behave correctly in most cases and avoid corrupting internal state.			
However, in the case of MemoryError in particular (perhaps others?) it is quite possible for an exception to corrupt internal state, leading to incorrect/unpredictable results and deadlocks when handling concurrent/subsequent fuse operations.			
This seems to be an inherent problem with the practice of using exceptions to handle routine errors. It's hard for an outer handler to know the difference between a routine error (passed up the stack gracefully by lower levels after cleaning themselves up with "finally" etc) and a real problem that has already caused the lower levels to abandon all hope of correctness. In the case of MemoryError specifically, which so many innocuous Python statements can encounter, it seems too optimistic to assume that it has been handled correctly by all inner layers without undefined side effects.			
If we want to return errors instead of crashing on MemoryError in the most obvious place (i.e., allocating a buffer to read/write Keep data) it would be safer to catch MemoryError <i>at the moment of allocation</i> and re-raise it as a less generic error that can be caught by the generic arv-mount "I/O error" code. That way, when <i>other</i> MemoryError exceptions happen (e.g., while allocating a dict) we can do the safe thing and crash out, rather than proceeding with undefined results.			

### History

#2 - 07/06/2021 09:08 PM - Peter Amstutz

- Target version deleted (To Be Groomed)