# LOOKING TO DO A PROJECT ON LINUX MEMORY OR PROCESS MANAGEMENT

KATRINA SIEGFRIED

```
__________ modifier_ob___
 mirror object to mirror
mirror_mod.mirror_object
peration == "MIRROR_X":
irror_mod.use_x = True
irror_mod.use_y = False
irror_mod.use_z = False
 _operation == "MIRROR_Y"
 Irror_mod.use_x = False
 lrror_mod.use_y = True
 irror_mod.use_z = False
  _operation == "MIRROR_Z";
  rror_mod.use_x = False
  rror_mod.use_y = False
  rror mod.use z = True
  melection at the end -add
    ob.select= 1
   er ob.select=1
   ntext.scene.objects.action
   "Selected" + str(modified
    irror ob.select = 0
  bpy.context.selected obj
   ata.objects[one.name].se
  wint("please select exaction
  OPERATOR CLASSES ----
  ypes.Operator):
   X mirror to the selecter
ject.mirror_mirror_x"
 ontext):
    object is not
```



## EMBEDDED AND SAFETY CRITICAL SYSTEMS

- Limited memory
- Real time scheduling
- Containerization used to segment processes

### AREAS OF INTEREST

- Current OOM Killer heuristic is not always predictable
  - New heuristics that more aptly detects a memory leak in constrained memory application
- Middleware to mark processes as sacrificial lambs to permit graceful shutdown
  - Medical device alert user to failure and safely power down (ex. Devices pumping blood need to slowly stop pumping, insulin pumps need to shut valves)
  - Automotive alert user and permit safety critical processes to move user out of harms way before impending shutdown
  - Sacrificial containers? Is it OK to lose UI or non-safety critical process if there is runaway mem allocation in safety critical container?
- Using thrashing to predict potential memory issues when memory is limited
  - Can you see the storm brewing?

- These are just my interests
- I'm flexible on a project idea
- Looking to learn!

Katrina.Siegfried@Colorado.edu

#### HALF BAKED, LOOKING FOR FULLY-BAKED

#### PERFORMANCE SURVEY ACROSS EMBEDDED OS

- Examine across three different devices
  - Use case analysis
  - Replicate / Emulate
- Memory Management
  - Profile / Benchmarks
  - OOM Killer induce memory leaks (std way of approaching that)
- Future Improvements
  - Proposed improvements to OOM
    - Protections to user-space processes
    - Safe-shutdown state (ciritical only mode)
    - Adaptive OOM Killer