- **Using Continuations to Implement Thread Management and Communication in Operating Systems Richard P. Draves, Brian N. Bershad, Richard F. Rashid, Randall W. Dean. SOSP '91**
  - **Mach paper – precedes the events and threads paper**

- **Manage thread stack inside and kernel by using application defined continuations**
  - **Can be optimized further**

Programming models

- ## Process model
  - One stack for every thread in the kernel

- ## Interrupt or event model
  - Uses a single, per processor stack

- ## When the paper was written, workstations had 32 MB main memory and laptops had < 8 MB
  - Why should we worry about saving stack space?

Prior solutions

- via application level threads (C-threads)

- Observation: multiprocessor programs tend to stay away from the kernel, multithreaded programs are kernel intensive

- Use interrupt model to redesign Mach
  - History of mach and other realities make this hard

Continuations

Before Continuations

```
/* a frequently used system call */

example(arg1, arg2) {
  P1(arg1, arg2);
  if (need_to_block)  {
    /* use process model */
    thread_block();
    P2(arg1);
  } else  {
    P3();
  }
  /* return control to user */
  return SUCCESS;
}
```

After Continuations

```
example(arg1, arg2) {
  P1(arg1, arg2);
  if (need_to_block)  {
    /* use continuation */
    save context in thread;
    thread_block(example_continue);
    /*NOTREACHED*/
  } else  {
    P3();
  }
  /* return control to user */
  thread_syscall_return(SUCCESS);
}

example_continue() {
  recover context from thread;
  P2(recovered arg1);

  /* return control to user */
  thread_syscall_return(SUCCESS);
}
```

Figure 1: Transforming a Blocking Kernel Procedure

Using continuations

- Support both types of threads
- Target problem areas and convert them to use continuations
  - Very small of points are problematic (6 of 60)

- Useful for stack discarding, stack handoff and continuation recognition,
- cross address space IPC
- Exception handling
- Preemptive scheduling – waiting threads continue in use space

Performance

- Time saved, space saved

- Software engineering concerns: would this kernel be a nightmare to maintain

- Question: how is this relevant for different class of modern machines?