```
processType processDescriptor[maxProcs];
int executing = 0;    # index of the executing process
declarations of variables for the free, ready, and waiting lists;
SVC_Handler: {   # entered with interrupts inhibited
  save state of executing;
  determine which primitive was invoked, then call it;
}
Timer_Handler: {   # entered with interrupts inhibited
  insert descriptor of executing at end of ready list;
  executing = 0;
  dispatcher();
}
procedure fork(initial process state) {
  remove a descriptor from the free list and initialize it;
  insert the descriptor on the end of the ready list;
  dispatcher();
}
procedure quit() {
  record that executing has quit;
  insert descriptor of executing at end of free list;
  executing = 0;
  if (parent process is waiting for this child) {
    remove parent from the waiting list; put parent on the ready list; }
  dispatcher();
}
procedure join(name of child process) {
  if (child has not yet quit) {
    put the descriptor of executing on the waiting list;
    executing = 0;
  }
  dispatcher();
}
procedure dispatcher() {
  if (executing == 0) { # current process blocked or quit
    remove descriptor from front of ready list;
    set executing to point to it;
  }
  start the interval timer;
  load state of executing;    # with interrupts enabled
}
```

Figure 6.2 Outline of a single-processor kernel.