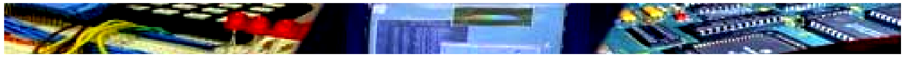
Embedded Systems Programming

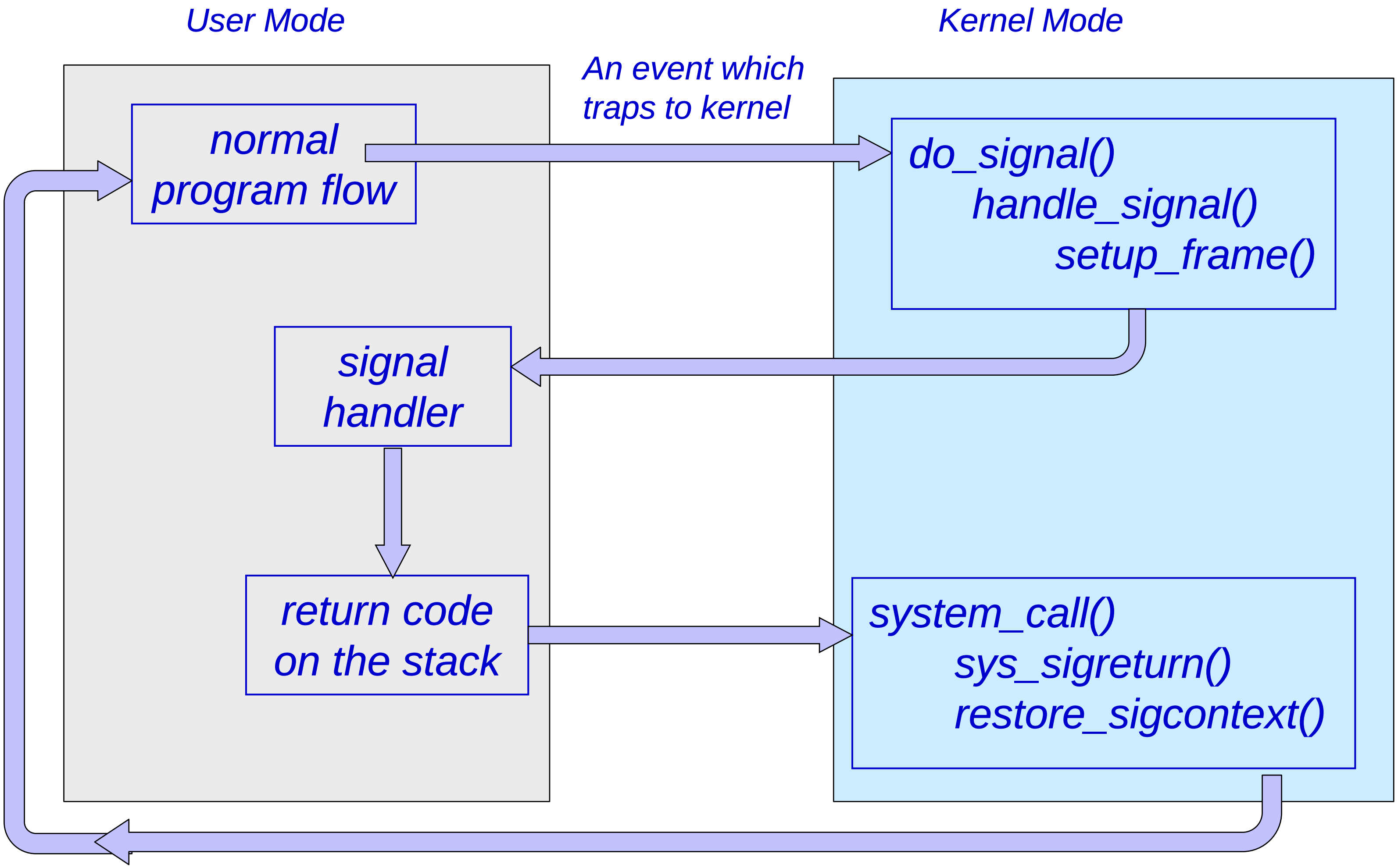
Kernel Signal Mechanism
(Module 25)

Yann-Hang Lee
Arizona State University
yhlee@asu.edu
(480) 727-7507

Summer 2014

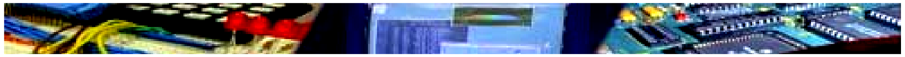


Scheme of Signal Processing



Signaling Process

- ***Posting* – action taken when event occurs that process needs to be notified of (signal generation)**
  - sending a signal – must be done in kernel mode, i.e. *send_signal, kill_proc,* etc.
- ***Delivery* – action taken when process recognizes arrival of event (signal handling)**
  - Before a process resumes execution in user mode, kernel checks for pending non-blocked signals for it. If yes, *do_signal*
- ***Catching* – if user-level signal handler is invoked, process is said to catch the signal**
  - in kernel, invoke *handle_signa*l and *setup_frame*
  - Process first executes signal handler in user mode before resuming "normal" execution.
- ***Pending* – signals that have been posted, but not yet delivered**

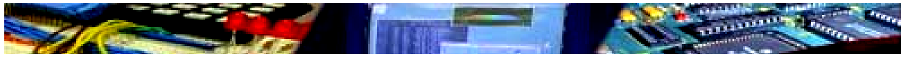


Kernel Data Structures for Signals

sigset_t : array of signals sent to a process

```
struct sigaction {
    void (*sa_handler)();     /* handler address, or  SIG_IGN, or SIG_DFL */
    sigset_t sa_mask;         /* blocked signal list */
    int sa_flags;             /* options e.g., SA_RESTART */
    }
```

```
struct task_struct { ……      /* for tracking sent, blocked and pending signals */
    struct signal_struct *signal;        /* signal descriptor */
    struct sighand_struct *sighand;  /*signal handler descriptor */
    sigset_t blocked, real_blocked;
    sigset_t saved_sigmask; /* restored if set_restore_sigmask() */
    struct sigpending pending;
    ……
    }
```



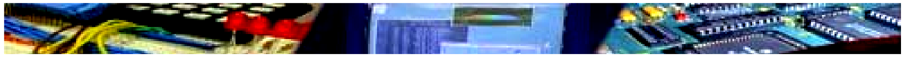
Real-time System Specification

❑ Logical correctness requirements:

- The computation produces correct outputs.
- Models of computation to describe inputs and computations
- Additional requirements on resource, security, reliability, etc.
- Finite state machine
  - good for control logic and protocols,
  - transition and activity
- Data flow – modular computations that are triggered by the availability of input data.

❑ Temporal correctness requirements:

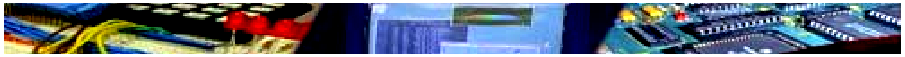
- The computation produces outputs at the right time
- When the computation can get started and should be completed



Specification Patterns

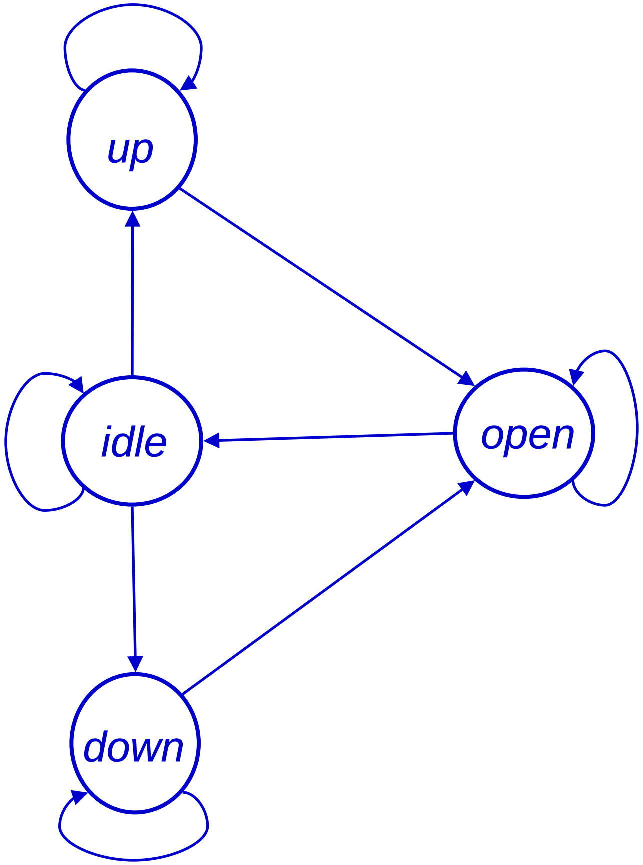
Category	Pattern	Example
Duration (stimuli and responses)	minimum duration	The system has a minimum 'off' period of 120 seconds before it reenters the cranking mode.
	maximum duration	The system can only operate in engine cranking mode for no longer than 10 seconds at one time
Periodic	bounded recurrence	The ABS controller checks for wheel skidding every 10 milliseconds.")
Real-time order	bounded response	The detection of and response to rapid deceleration must occur within 0.015 seconds.
	bounded invariance	If Error 502 is received, then the braking system is inhibited for 10 seconds.

(S. Konrad and B.H.C. Cheng “Real-time specification patterns”, ICSE 2005)



RT Specification in FSM

- ❑ Duration of staying in a state
- ❑ Periodic activity in a state
- ❑ Bounded response for each transition
- ❑ Accumulated delay between multiple transitions
- ❑ Hierarchical FSM
  - ❖ a state encloses a FSM
  - ❖ enter a state → activate a FSM
- ❑ Concurrent FSM
  - ❖ FSMs run in parallel (active simultaneously)


up
idle
open
down