Exceptions

Normal flow control
1. next instruction immediately follows
2. " " rest of jump branch call

Exception flow control
1. external event (e.g., interrupt)
2. internal event (e.g., page fault, div by zero)
3. explicit request (e.g., ask for control block removal)

New Areas

OS

Explicit tables of pointers to exception handlers

- Code for exce. 0
- Handler for exce. 1
Interrupts

Action:
- Complete curr inst.
- Branch to interrupted int. service routine (ISR)
- Return to next inst.

Hints when writing ISRs:
- Keep it short
- Queue important work for later

Faults

Action:
- Branch to except handler
- Handler decides whether to abort or return to curr inst.

Abort
- Aborts terminate process

Trap

Action:
- Branch
- OS provides required service
- Ret. to next inst.

Processes

Process is an instance of a running program.

Logical control flow:
- The process has a private address space.
- The process can have multiple threads running concurrently.
fork
int fork (void)

if (fork () == 0)
    printf(" child ")
else {
    printf(" parent 

void exit (int status)

parent maps status vs. wait ()

exit()