the definition of pure micro-kernels: kernels that provide only address spaces, threads and IPC, or an equivalent set of primitives

L4: a second-generation microkernel

L4 features

message-based synchronous IPC(inter-process communication)

external paging mechanism

security mechanism based on secure domains

\tip the benefits of microkernel over monolithic: more secure, simpler, extensible, smaller with fewer bugs;

the drawbacks of microkernels -> since not everything in the kernel, more context switch -> lower performance

e.g. the need for extensibility example: different paging algorithm

### L4 Essentials

two basic concepts: threads and address spaces.

address space -> physical address

user-level recursive addressing mechanism can be built.(like virtual memory, called pager))

hardware interrupt: IPC implementation

exceptions and traps: only mirrors to the user level

### Linux Essentials and the transportation

\tip: the construction of Linux on L4 -> the system calls in Linux are re-complied/ linked with a library/ dynamic change the binary code on runtime to be suitable in L4.

Linux features

process and resource management

file systems

networking subsystems

device drivers

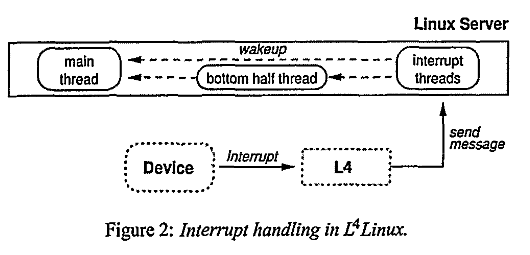
three-level architecture page-tables -> one-layer architecture-dependent macro

interrupt handlers: tip halves and bottom halves

memory mapping: one-to-one physical address mapping

only a single L4 thread, multi-thread implemented in Linux server

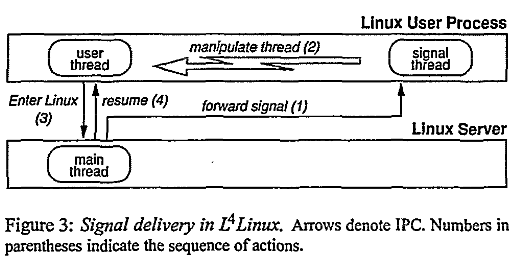
linux interrupt handler: threads waiting for the interrupt message from L4



linux user process - an L4 task - an address space with a set of threads

system calls: remote procedure calls

singal:



scheduling is done by the L4 internal scheduler

Inside the L4 - priorities with round robin

only one address space per LInux user process, the server space is not replicated.

(instead of the dual-space approach) - 没有看懂这里，什么是dual-space appraoch?D

### experiments and results

three questions and the corresponding experiments:

* comparison of L4Linux with the pure LInux/x86
  + L4Linux with around 5% to 10% penalty
* comparison of different underlying L4 microkernels
  + matters, MkLinux worse
* comparison of the L4Linux ot the in=kernel version of MkLinux to find how much the colocation improve performance
  + not enough to cover the performance gap of the underlying microkernel

two kinds of benchmarks:

microbenchmarks

macrobenchmarks

### the plus performance L4Linux offers

specialization - improve the existed performance

extensibility - provide those services which are not existed in the previous systems