Caching and Threads

COS 461

Muneeb Ali
Caching

- Why cache?
- How to cache?
Caching

- On-disk Cache

- On-disk + In-memory Index

- On-disk + In-memory Index + In-memory Cache
Threads

- Why threads?
Threads

- Why not just fork()?
Threads

- Why not just fork()?
Threads

Typical process address space

Virtual address space

Program code (text)

Static data

Heap (dynamically allocated)

Stack (dynamically allocated)
Threads

- Process defines address space
- Threads share address space

Process Control Block (PCB)
- process-specific info
- PID, owner, heap pointer, active threads
  and pointers to thread info, files

Thread Control Block (TCB)
- thread-specific info
- stack pointer, PC, thread state,
  register …

Process address space (with threads)
Threads

(single-threaded process)

(multithreaded process)
Pthreads - basics

- Less overhead for creating a thread (~ 2 milliseconds)

- IEEE's POSIX Threads Model (Standard for API)

- Compile: -pthread

#include <pthread.h>

pthread_t threads[NUM_THREADS];

rc = pthread_create(&threads[t], NULL, PrintHello, (void *)t);

void *PrintHello(void *threadid)
{
    ....
    pthread_exit(NULL);
}
```c
pthread_attr_t attr;
pthread_attr_init(&attr);
pthread_attr_setdetachstate(&attr, PTHREAD_CREATE_JOINABLE);

rc = pthread_create(&thread[t], &attr, BusyWork, (void *)t);
rc = pthread_join(thread[t], &status);
```
Demo

- Mutex

- Write-write conflict

Image source: ZDNet
Optional Material

- `select()` system call

- `O_NONBLOCK` socket option
Optional Material

- select() system call

- O_NONBLOCK socket option

Threads vs. Events debate

- Age old debate – (late) Roger Needham duality argument 1978

- Threads are a bad idea! (John Ousterhout USENIX’1996)

- Events are a bad idea! (Eric Brewer HotOS’2003)

- Protothreads (SenSys’06)
Questions?

Thank you!
Happy thread-ing!