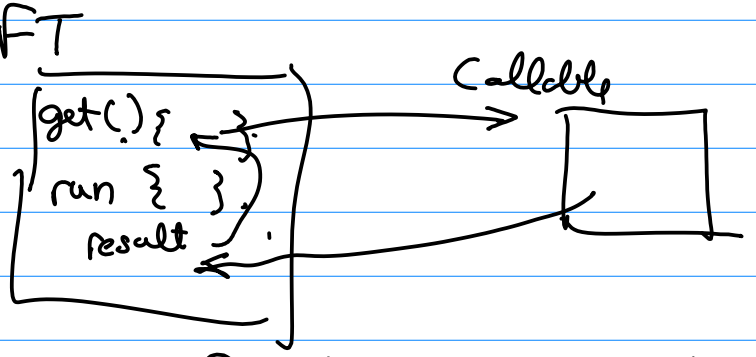
CptS 483/580 Feb. 14, 2011

Future Task — objects represent computations that can be run synchronously or asynchronously and once complete can deliver a result.

- Callable -



ft. get() — waits for the result to be available

```
ft = new FutureTask ( some Callable )
t = new Thread(ft)                    t.start.
```
pass the ft off to a Thread constructor

```
res = ft.get();
```

---

end of Ch-5

Goal: a cache of results of calling an expensive function on particular arguments

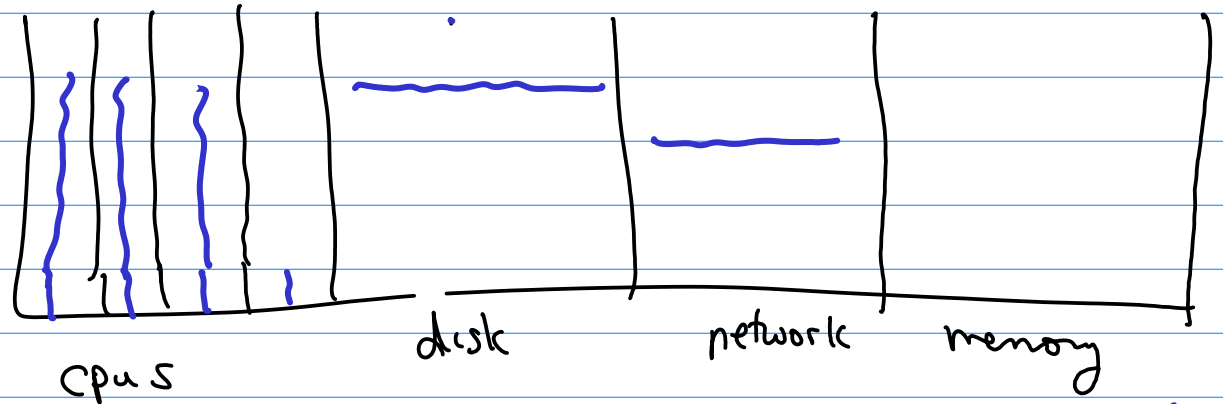
f(a1) .. f(a2) ... f(a1)

↳ remember results of first use with each argument

How to do this in a thread-safe and thread-efficient way?

Chapt. 6: Running the "right" number of threads. —

Executors



Thrashing

So to avoid creating too many threads, Java library offers Executor model:

- as app programmers we think in terms of tasks to be done rather than threads.
  - compute this
  - fetch a URL
  - fetch a disk block
  - write a file

Executors are objects that make a policy-based decision about <u>how</u> to execute a task.

- execute in a new Thread?
- " " " Thread Pool?
- " " the current task.

executor.<u>execute</u>(Runnable cmd)

Some provided executors are:

independent tasks
- Fixed Thread Pool — a fixed # of threads.
- Cached Thread Pool - no fixed maximum # of threads

sequentially dependent
- Single Thread Executor - guarantees that later tasks see results (side effects) of earlier tasks.

When are executors no help?

Consider Piplie with a blocking queue between two threads.

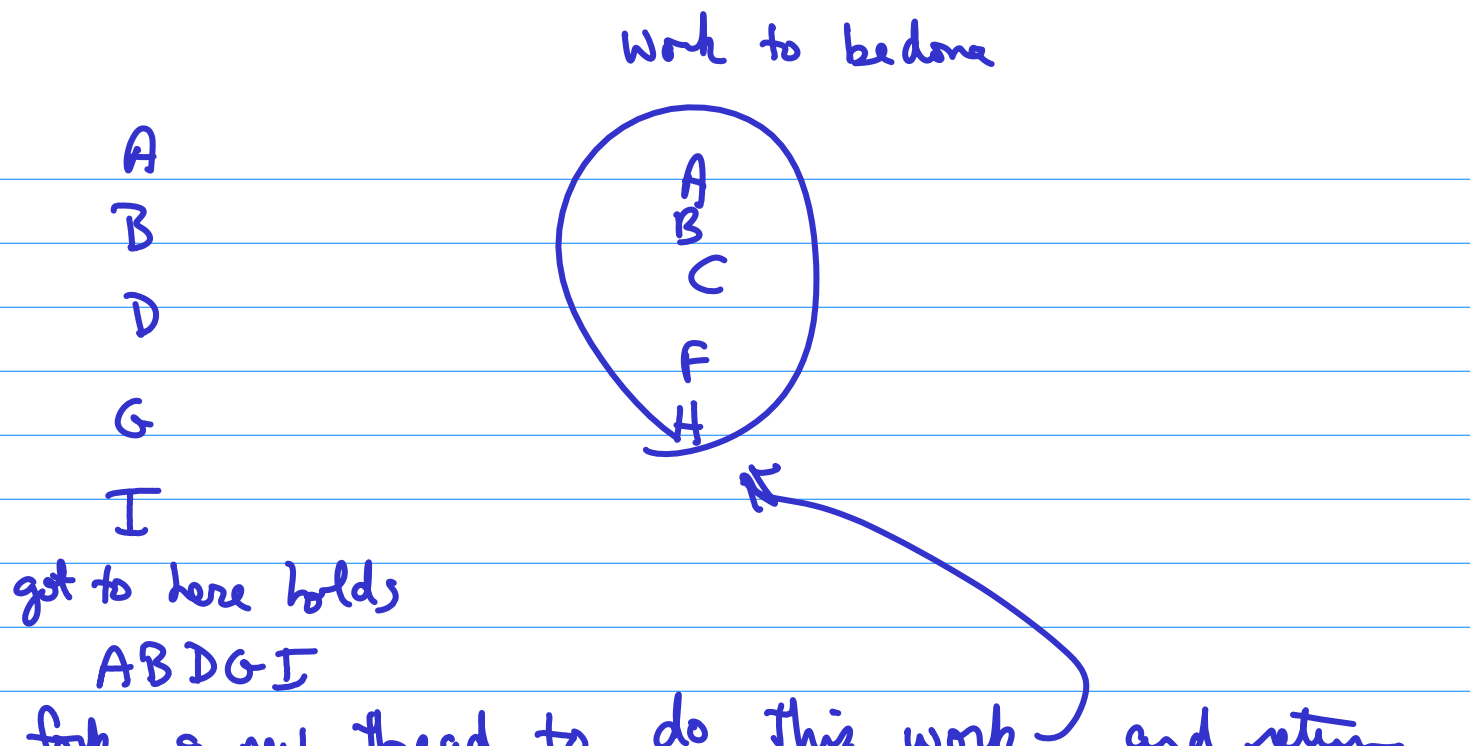
Uses of threads:

- Defer work — good application of executor
- Periodic work — sleeps a while, computes a while — Cron
- scheduled future work — at
- exploit parallelism (1% of # of threads)
- deadlock avoidance.
  - to avoid deadlock, it is sufficient to always take locks in the same order in all threads.

T1	T2
A	A
B	C
	B

T1	T2
A	B
B - blocked	A - blocked

Deadlock.


Work to be done
A
B
D
G
I
A
B
C
F
H
got to here holds
ABDGI
fork a new thread to do this work and return.
Pipelines
T1 → Q1 → T2 → Q2 → ....
Bounded Buffer
Input Devices
I1
I2
I3
Q1
T1
Slack Thread