Blocking, Polling, & Utilization

- Learning Objectives:
  - Explain the difference between blocking and polling.
  - Discuss the tradeoffs between blocking and polling.
  - Be able to use utilization as a metric to measure system efficiency.

Two Uses of `waitpid`

- In the fork/exec examples, we demonstrated how to use waitpid to allow a parent to wait for its children to exit.
- Without any flag values, specified, `waitpid` is a blocking system call.
  - That means that it blocks the current process from doing any further work until something happens.
- Some of you stumbled upon what happens if you specify WNOHANG.
  - "The WNOHANG option is used to indicate that the call should not block if there are no processes that wish to report status."

Compare and Contrast

Blocking waitpid:

```
p = fork();
waitpid(p, &status, 0);
```

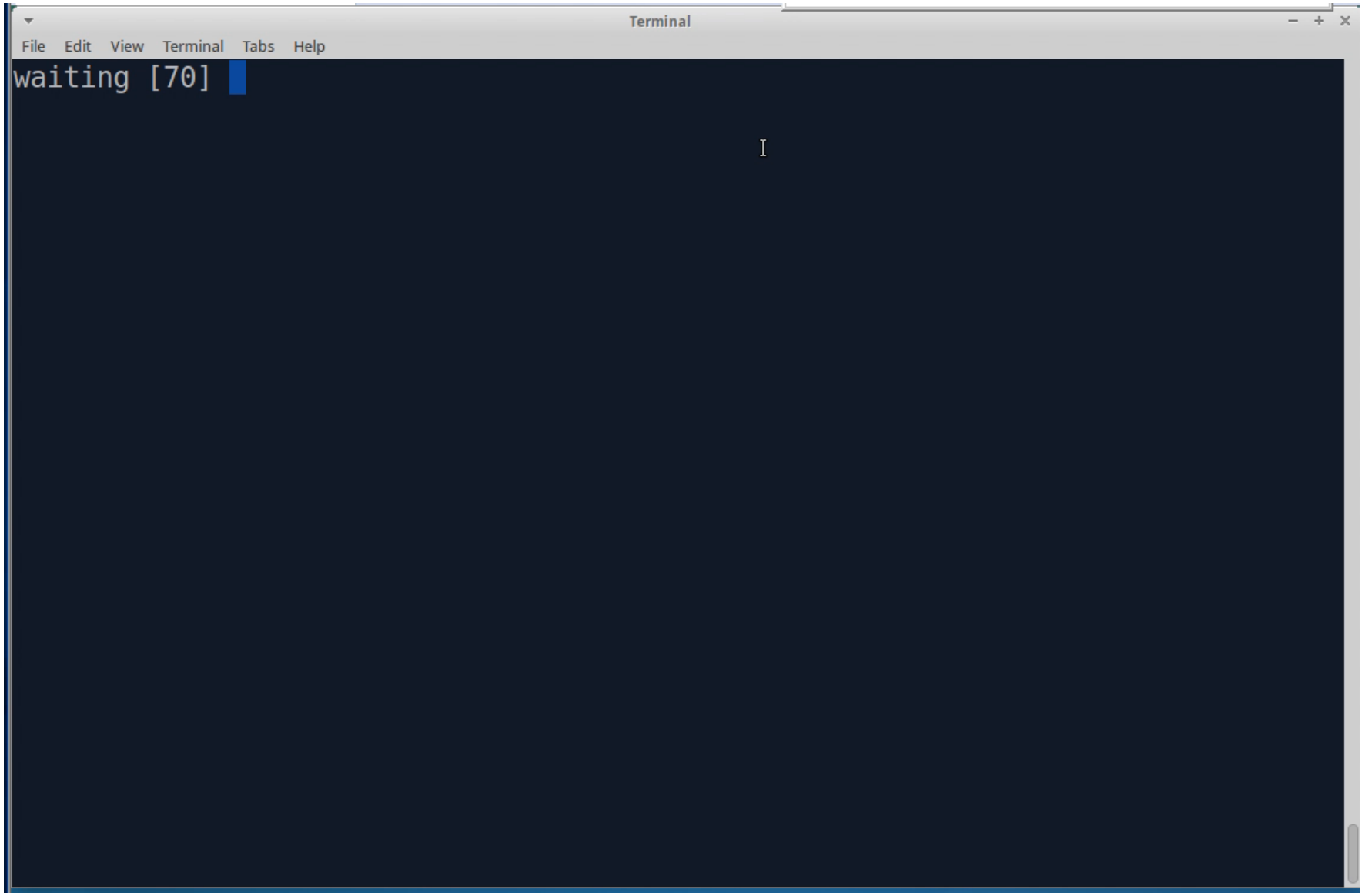
Polling waitpid:

```
p = fork();
while (p != waitpid(p, &status, WNOHANG));
```

What is the difference between these two uses?

Measuring Efficiency

- How do you measure how efficiently something is being done?
- We want to use utilization as the metric.
  - The fraction of the processor being used to do useful work.
- However, you will see utilization used in different ways:
  - "The processor was 100% utilized."
  - "We were able to do thus with only 1% processor utilization."
- Are big numbers good or bad?


Terminal
File Edit View Terminal Tabs Help
waiting [70]

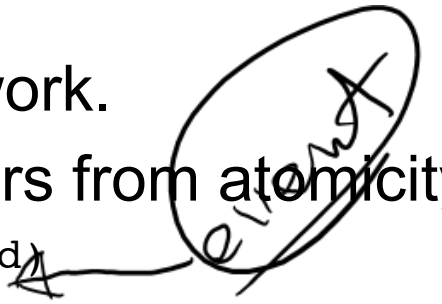
Interpreting Utilization

- If you can differentiate useful work from wasted work and express utilization in terms of useful work, then you want BIG numbers.
  - "We achieved 98% utilization."
- If you can't differentiate useful work from wasted work, then you'd like to accomplish as much work as possible with the processor reporting the lowest utilization.
  - "We managed to keep processor utilization under 60% during the entire holiday season."
- Note: Most systems don't work well when they have any resource approaching 100% utilization, regardless of whether doing useful or wasted work.

Tradeoffs between Polling and Blocking

- Blocking avoids wasted work.
- Blocking sometimes suffers from atomicity problems:

```
if (event hasn't happened)
    issue blocking call
```



- Polling can sometimes be more responsive:
  - If it takes you longer to block than the time it takes for the event on which you are waiting to complete, polling might be better.
- We often refer to polling as busy-waiting (we keep the processor busy while we wait). In general, you should avoid busy-waiting!