Being a Marker

Julian Bradfield

What do staff want of a marker?

What do staff want of a marker?

- Understands the material. It is annoying having to re-mark things. It is strictly less annoying if the marker needs to ask us questions.

What do staff want of a marker?

- Understands the material. It is annoying having to re-mark things. It is strictly less annoying if the marker needs to ask us questions.
- **Meets deadlines.** Late return reflects badly on everybody. So you must schedule your marking time!

What do staff want of a marker?

- Understands the material. It is annoying having to re-mark things. It is strictly less annoying if the marker needs to ask us questions.
- **Meets deadlines.** Late return reflects badly on everybody. So you must schedule your marking time!
- Can follow a marking scheme. Even if you don't agree with it. But you may get a lot of discretion to invent the marking scheme, at least in fine details.

What do students want of a marker?

What do students want of a marker?

- Understands the material. It may be nice to know they're cleverer than the marker, but ...

What do students want of a marker?

- Understands the material. It may be nice to know they're cleverer than the marker, but ...
- **Meets deadlines.** Many students don't even look at their feedback. Those who do, really care.

What do students want of a marker?

- Understands the material. It may be nice to know they're cleverer than the marker, but ...
- **Meets deadlines.** Many students don't even look at their feedback. Those who do, really care.
- Can follow a marking scheme. Students talk to their friends!

What do students want of a marker?

- Understands the material. It may be nice to know they're cleverer than the marker, but ...
- **Meets deadlines.** Many students don't even look at their feedback. Those who do, really care.
- Can follow a marking scheme. Students talk to their friends!
- **Gives helpful feedback.** Explains what doesn't work, why it doesn't work, and how it should work.

Do students want feedback?

They say they do.

Do students want feedback?

They say they do.

But, the ITO has piles of uncollected coursework at the end of semester.

Do students want feedback?

They say they do.

But, the ITO has piles of uncollected coursework at the end of semester.

A few years ago, I started recording audio feedback for things, including the mock exam for Computer Programming – 140 students, typically. (There is auto-marker feedback anyway, but it can be fairly cryptic.)

Do students want feedback?

They say they do.

But, the ITO has piles of uncollected coursework at the end of semester.

A few years ago, I started recording audio feedback for things, including the mock exam for Computer Programming – 140 students, typically. (There is auto-marker feedback anyway, but it can be fairly cryptic.)

Why? Audio is more personal, feels like much less work than typing stuff, and allows you to say more. It still takes 5–10 minutes per student.

Do students want feedback?

They say they do.

But, the ITO has piles of uncollected coursework at the end of semester.

A few years ago, I started recording audio feedback for things, including the mock exam for Computer Programming – 140 students, typically. (There is auto-marker feedback anyway, but it can be fairly cryptic.)

Why? Audio is more personal, feels like much less work than typing stuff, and allows you to say more. It still takes 5–10 minutes per student.

Last year, before doing it, I asked the class to tell me if they actually wanted feedback. How many do you think requested it?

Do students want feedback?

They say they do.

But, the ITO has piles of uncollected coursework at the end of semester.

A few years ago, I started recording audio feedback for things, including the mock exam for Computer Programming – 140 students, typically. (There is auto-marker feedback anyway, but it can be fairly cryptic.)

Why? Audio is more personal, feels like much less work than typing stuff, and allows you to say more. It still takes 5–10 minutes per student.

Last year, before doing it, I asked the class to tell me if they actually wanted feedback. How many do you think requested it?

So, much of your work may be wasted. But

- you're being paid for it;
- some students really do care and want your feedback.

Example

For several years, I taught the UG3 Operating Systems course. There was a practical exercise, which involved messing around with concurrency primitives inside a Linux kernel.

As I told the students, "you can do this practical by writing 15 lines of code . . . but finding the right 15 lines is the hard part".

Extract from the specification

Your task is to produce a module that starts a kernel thread
that takes action to alleviate this problem, by preventing
busy-waiting processes from executing while there appears to be use
activity going on. On the other hand, the busy-waiting process shou
not be completely starved: it should get some run time, even if
the user is typing continously. (The user will necessarily be
delayed while the busy process is running, but that's too bad.)

To keep things simple, you should stick with the worker scheduli
policy of just doing its work every ten seconds.
The worker's work should be as follows:
if a key has been pressed in the last ten seconds (i.e. since the
last time round), any excessively busy process should be prevented
from executing during the next minute (i.e. six times round).

Extract from the marking scheme

```
10pt for stopping and resuming tasks in some way
- stopping busy tasks : up to 4pt
- resuming tasks after 60sec : up to 6pt
- doing it in a nasty way : up to -3pt
- using send_sig or force_sig to send SIGCONT instead of kill_proc:
- possibility of starvation of the busy task : -2pt
- stopping the busy tasks too early after resume : -1pt
- not detecting if the busy tasks are already dead or woken up or
  already stopped : up to -2pt

8pt for implementing the two criteria for a busy task
- implementing only the first criteria : -5pt
- implementing only the second criteria : -3pt
- sth wrong with the condition eg. rounding error in dividing int
  numbers, || on the two conditions instead of && : up to -2pt
- using current_kernel_time() instead of
  do_posix_clock_monotonic_gettime(struct timespec*) : -1pt
- traversing the task list improperly : -3pt
```



```
2pt for having the module clean up properly ie. starting previously
```

Extract from general feedback

Many people had problems when writing correct criteria for being a busy task. Very common were rounding errors such as:

```
float perc;
int time_in_kernel, time_running;
perc = (time_in_kernel / time_running) * 100;
```

This will always assign 0.0 to the perc variable as time_in_kernel always less than time_running and the arithmetic operation (time_in_kernel / time_running) will be done in integers returning a value only the integer part of this operation without the reminder. Besides using floating point numbers inside the kernel is highly discouraged and almost always unnecessary, as it was in this case. Also some people used || instead of && to join the two conditions of being a busy task that made any task that spent more than one second in the kernel to be considered busy. Most of the ti there are many such tasks, but they shouldn't be considered busy.

Extract from individual feedback

```
Q3.

34pt
could stop busy tasks, resume tasks after 60sec, implement the two
criteria for a busy task, works for more than one busy process on a
UP machine, but it does not have the module clean up properly
ie. starting previously stopped busy tasks, not SMP safety(may thin
about blocking the task list).Additional 2pt for good coding style.
```

Another extract from individual feedback

```
Q3.

29pt
could stop busy tasks, resume tasks after 60sec, implement the two
criteria for a busy task, works for more than one busy process on a
UP machine. It has the module clean up properly, but not SMP
safety(may think about blocking the task list).Additional 2pt for g
coding style. The code has a bug: if the number of busy tasks is mo
than 6, the program will crash. You set "int pidsToKill[6]" but the
is not any boundary check while you fill the array.
```