



How to give a good presentation

Tony Field

With material from Simon Peyton Jones's
"Research Skills"

(<http://research.microsoft.com/~simonpj/papers/giving-a-talk/giving-a-talk.htm>)



Your project presentation...

- 40 minutes total, including any demo, and **at least** 10 minutes for questions
- Your chance to show off your achievements
- Our chance to find out what you really did
- Assessed by a team of academics to ensure accuracy and consistency
- Treat the presentation as an **opportunity** to tell us what you've achieved



The presentation...

- The presentation carries no marks in its own right
- It is **compulsory** – no presentation means no project mark
- The purpose is to ensure you get the **right mark**, NOT to mock you, your work or your presentation skills



The audience you would like...

- Have read every word of your report
- Have previously studied all 5,929,913 lines of code in the Linux kernel
- Are all agog to hear about your patch to allow interrupt timings from network devices to contribute to the kernel entropy pool
- Will willingly filter out the waffly, irrelevant parts of your talk and extract the “meat” for themselves



In practice, your audience...

- (Bar your 1st/2nd markers) Have **no idea** what your project is about
- Know that Linux has a kernel, but have no idea why you'd ever want to "patch" it
- Will focus their minds onto something else (research, football, opera...) if you don't hold their attention



Your mission is to...

**GET THEM
INTERESTED**

KEEP THEM INTERESTED

And make them glad they were there to
listen!



Some “do”s



Begin with the problem

The first **TWO** minutes of every talk is absolutely precious - you **must** engage the audience with your motivation

- What is the problem?
- Why is it an interesting problem?
- What's the crux of your idea?

So many talks fail at this first hurdle



Example messages

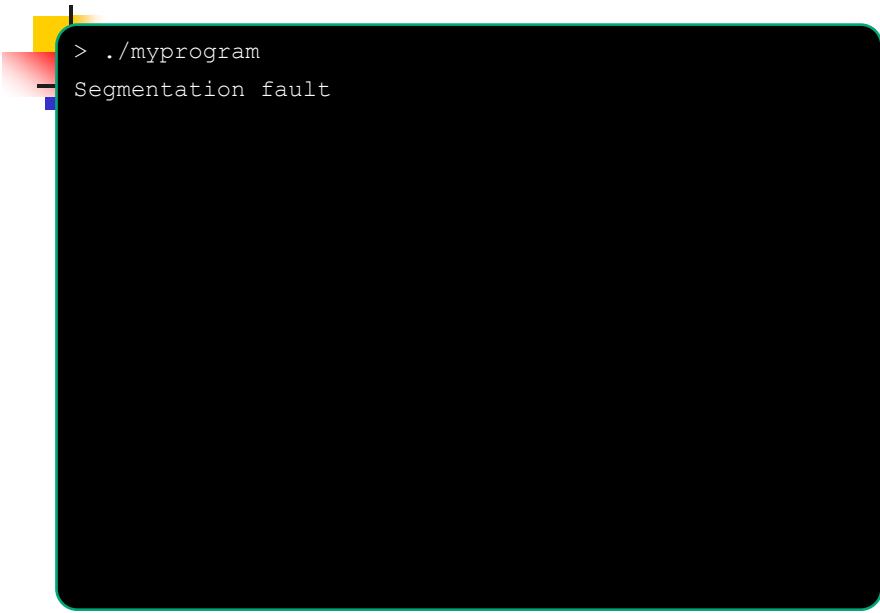
- When C programs access memory illegally it's very hard to uncover what happened (give example). Bounds checking can **really** help (give example). I'm going to show you how I built a state of the art bounds checker for gcc...
- Game physics engines sometimes glitch when there are large numbers of objects in the game (give example). My idea is to switch to simpler object representations when this happens, to speed up the physics. I've explored the idea by extending the Bullet physics engine...




Finding What C++ Lost:

Tracking Referent Objects with GCC

Alexander Lamaison

A terminal window with a black background and a green border. On the left side, there are three overlapping colored squares: yellow, red, and blue. The terminal shows a prompt character followed by the command `./myprogram`, and the next line shows the error message "Segmentation fault".

```
> ./myprogram
Segmentation fault
```

A terminal window with a black background and a green border. On the left side, there are three overlapping colored squares: yellow, red, and blue. The terminal shows a prompt character followed by the command `./myprogram`, and the next line shows another prompt character.

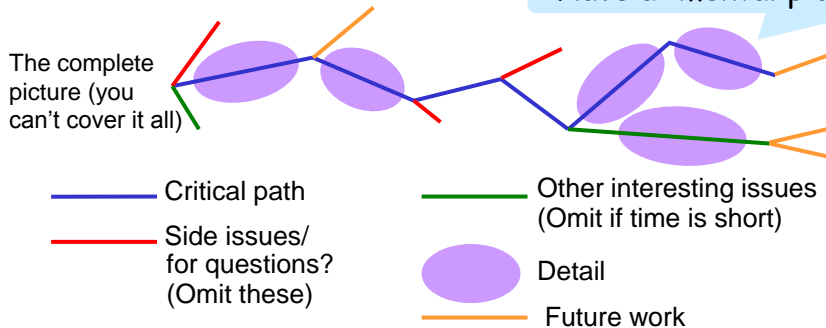
```
> ./myprogram
>
```

```
> ./myprogram
*****
bounds violation caused by earlier pointer arithmetic:
The arithmetic took place at myprogram.cpp:5 (main)
The violation occurred at myprogram.cpp:6 (main)
Before going out-of-bounds the pointer referred to
    myprogram.cpp:3 (main) a
time=1181317406.431297 ptr=0xbfaf12cc size=4
pc=0xb7f57124
```

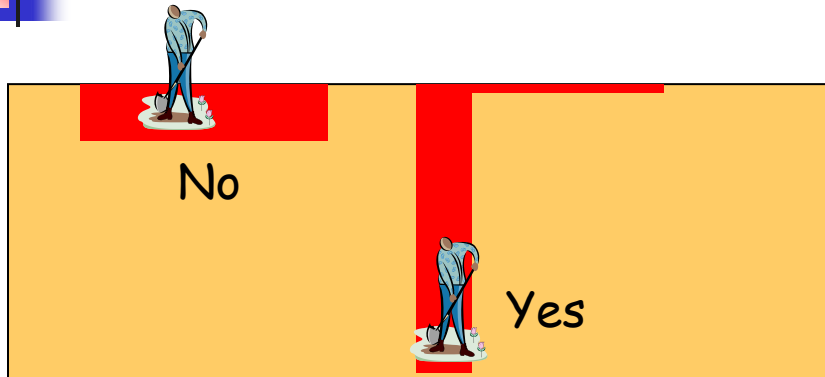
Make a plan for your talk

- You cannot (hopefully!) explain everything you have done
 - Try to tell a coherent story from **end-to-end**
 - Ruthlessly **prune** material that is not on the critical path
 - Focus** on the interesting detail

Have a "mental picture"



Narrow, deep beats wide, shallow



Avoid shallow overviews

Cut to the chase: the technical “meat”

Use examples and diagrams

- **Examples** and **diagrams** are key weapons – use them in place of text whenever possible
- Some of the best talks have little or **no** linking text at all

REMARK: I am using **lots** of text here so you can read the slides later on!



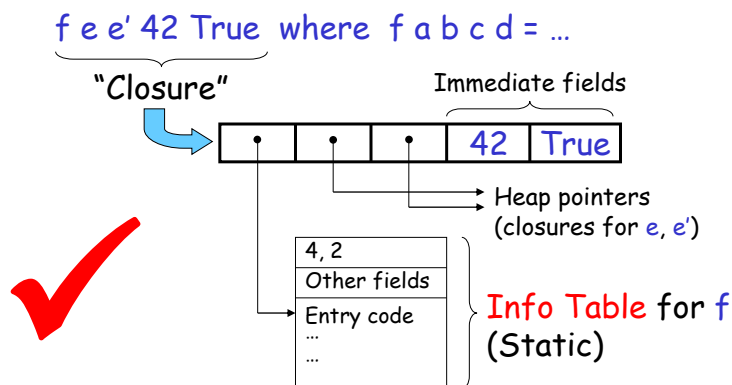
Use examples and diagrams

Closures in GHC

- Closures are used to represent unevaluated expressions
- They are stored on the heap
- The first field points to the 'info table'
- The remaining fields can be either
 - Pointers to other heap objects
 - Immediate (literal) values



Use examples and diagrams





Some “don’t”s



Don’t give an “Outline of my talk”

- What are fluid queues?
- The Boxma-Dumas model
- Limitations of the Boxma-Dumas model
- Laplace Transform of the busy period distribution with multiple emptying states
- Semi-numerical solution
- Results
- Conclusions and further work





No outline!

An “Outline of my talk”: conveys **near zero** information at the start

- But you might highlight key issues that your talk will address, **after** your introduction...
- ...with optional signposts at pause points during the talk



Don't give a literature survey

- Time is precious – the survey is in the report
- But **DO** make it clear that you know your background:
 - Respond readily to questions
 - Summarise key existing solutions, techniques etc. and their pros/cons
 - Mention relevant related work as you go along
 - Do not disparage the “opposition”



Don't use reams of text...

- A temptation is to write out everything you want to say on the slides or at least have lots of text to jog your memory
- This will certainly mean you won't miss anything
- However, the audience will not read a single word
- Instead they will listen to you as you read out each bit. For a while...
- It will sound very dull and very boring
- Then the audience will think... "I could read this talk later by looking at the slides. I know, I'll do a bit of work on that fluid queue paper I'm writing..."
- Meanwhile, you have some problems:
 - Do you read the text **exactly** as it appears?
 - Or do try to add a few extra words to make it sound more interesting?
 - Or do you skip some of the words, knowing that the audience can fill in the gaps for themselves?
- After a while, you begin thinking to yourself "Hmmm - This isn't going very well"
- Then, just as you are getting to the end of the slide (phew!), you realise...

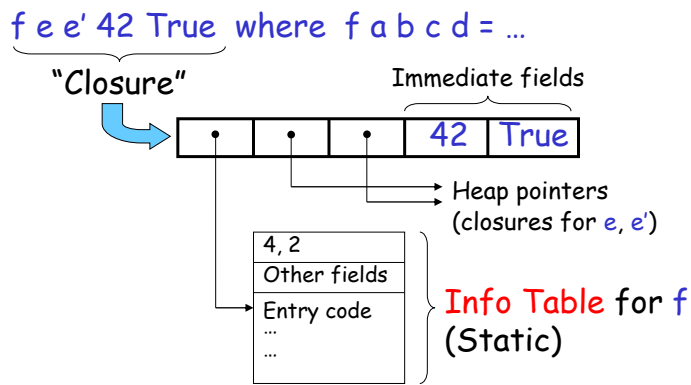


Don't use reams of text...

- That there is another of these 'monsters' on the next slide!
- And the next slide, and the next slide, and the next slide...
- Now you have a dilemma
 - Do you talk through the slides much quicker?
 - Do you start skipping over some points, saying things like "I think that point is pretty obvious", "I probably don't need to say that"...
- Meanwhile, that fluid queue paper is now out and the eyes are down; pens in hands
- Anyone lucky enough to bring a laptop is now doing their email
- The sound of tapping keys and pages turning takes its toll
- You start sweating – pages of text fly by as you stab the 'PgDn' key...
- You run out of excuses as to why you don't need to stop and read out each slide as it flies by
- And then... A revelation...
- You remember that somewhere... later... there's that DIAGRAM...!



Phew...!



NOW... the audience looks up, the tapping stops...



Avoid excessive technical detail

- Dense clouds of notation will send your audience to sleep
- Present specific aspects only; refer to the report for missing details
- **BUT**... have backup slides to use in response to questions





Technical detail

Sometimes it's useful to `see' some detail, but don't try to explain it all

$$\begin{array}{c}
 \frac{}{\Gamma \vdash k : \tau_k} \quad \frac{\Gamma \cup \{x : \tau\} \vdash e : \tau'}{\Gamma \vdash \lambda x. e : \tau \rightarrow \tau'} \quad \frac{\Gamma \vdash e_1 : \text{ST } \tau^\circ \tau \quad \Gamma \vdash e_2 : \tau \rightarrow \text{ST } \tau^\circ \tau'}{\Gamma \vdash e_1 \gg e_2 : \text{ST } \tau^\circ \tau'} \\
 \\
 \frac{\Gamma \vdash e : \tau}{\Gamma \vdash \text{returnST } e : \text{ST } \tau^\circ \tau} \quad \frac{\Gamma \vdash e : \tau}{\Gamma \vdash \text{newVar } e : \text{ST } \tau^\circ (\text{MutVar } \tau^\circ \tau)} \quad \frac{\Gamma \vdash e : \text{MutVar } \tau^\circ \tau}{\Gamma \vdash \text{readVar } e : \text{ST } \tau^\circ \tau} \\
 \\
 \frac{\Gamma \vdash e_1 : \text{MutVar } \tau^\circ \tau \quad \Gamma \vdash e_2 : \tau}{\Gamma \vdash \text{writeVar } e_1 \ e_2 : \text{ST } \tau^\circ \text{Unit}} \quad \frac{}{\Gamma \cup \{x : \forall \alpha_i. \tau\} \vdash x : \tau[\tau_i / \alpha_i]} \\
 \\
 \frac{\Gamma \vdash e : \tau' \rightarrow \tau \quad \Gamma \vdash e' : \tau'}{\Gamma \vdash e \ e' : \tau} \quad \frac{\Gamma \vdash e : \text{ST } \alpha^\circ \tau \quad \alpha^\circ \notin FV(\Gamma, \tau)}{\Gamma \vdash \text{runST } e : \tau} \\
 \\
 \frac{\forall j. \Gamma \cup \{x_i : \tau_i\}_i \vdash e_j : \tau_j \quad \Gamma \cup \{x_i : \forall \alpha_{j_i}. \tau_i\}_i \vdash e' : \tau'}{\Gamma \vdash \text{let } \{x_i = e_i\}_i \text{ in } e' : \tau'} \quad \alpha_{j_i} \in FV(\tau_i) - FV(\Gamma)
 \end{array}$$

Figure 1. Typing Rules



Presenting your talk





How to present your talk

By far the most important thing is to

be enthusiastic



Nerves

You do not need to be nervous. But if you are...

- Remember:
 - The audience is **not** examining your presentation skills
 - You know something the audience doesn't!
 - The audience is **on your side** – they want to find out what you've done
- Script your first few sentences precisely; getting going is often the hardest part



Being seen, being heard...

- Avoid monotone 'stand and deliver'
- Move around, change the focus of attention, use the whiteboard
- Point at the **screen**, not at the overhead projector, or in mid air
- Don't use a laser pointer!

NB For a larger audience...

- Make sure you can be heard at the back
- Repeat questions so that everyone can hear



Hiding your point

A very annoying technique

- is to reveal
- your points
- one
- by one
- by one, unless...
- there is a punch line





Finishing

Absolutely without fail,
finish on time

- Audiences get restive and essentially **stop listening** when your time is up. Continuing is very counter productive
- Simply truncate and conclude
- **DO NOT** try to rescue the situation by speeding up



Beyond this talk...

The general standard of
presentation is **so low**
that you don't have to
be outstanding to stand
out!

You will attend 50x as many talks as you give. Watch other people's talks intelligently, and pick up ideas for what to do and what to avoid.