NEWS FROM COMPUTER SCIENCE AND ENGINEERING

Written by the Beta Team of CSESoc
Produced by Wen Di LIM
Edited by CSESoc EXEC, Tim Wiley
and Samuel POLLOCK.

βETA

Free as in speech and our awesome BBQs.

Beta Issue # 82
Iota edition
Week 04 Session 2, 2013

TECH

Cubesats

When I say ’satellite’, most people will think of a box in the sky sporting a large dish and a pair of photovoltaic wings. Others may think of the Hubble Space Telescope, Google Maps, or phone calls from boats. All will think of them as expensive - the domain of governments and large corporations. Historically, this has been the case, but within the past decade, things have started to change.

In the late 1990s, professors from Cal Poly and Stanford got together and came up with something amazing: a design containing all the subsystems of a satellite but small and cheap enough for a team of students to build, test, and operate. Of course, the technology of the time meant that they wouldn’t be able to do much more than make a few basic measurements and beep but as they were teaching tools, this was OK.

The key to this breakthrough was standardisation. Each satellite would be shaped like a stack of at most three 10 by 10 by 10 centimetre cubes, weigh up to a kilogram, and be nestable into a P-POD, a tube with a spring in the bottom able to store these satellites during launch and spit them out afterward when in orbit.

The first six cubesats launched in 2003 aboard a reworked ICBM. More followed in 2005, and since then their relatively low cost and quick build times have led to ever increasing numbers taking to the skies. This year, 19 have already entered orbit with a further 62 planned for later in the year. In total, cubesats will make up nearly 25% of all satellites launched in 2013.

What makes cubesats marvellous is their versatility. Cubesat missions have tracked ships, looked for earthquakes, hosted biological experiments, flown in formation, tested new technologies, broadcasted the Ecuadorian national anthem, and studied space weather. They have sported tethers, solar sails, and plasma thrusters. OSSI-1 is an open-source artistic meditation on fantasy and reality, inviting ordinary people to connect with the heavens. STraND-1 and PhoneSat 1.0 have Google Nexus Ones for brains.

Cubesats are even pioneering business models: ArduSat paid for itself through a Kickstarter campaign which gave pledgers the chance to take pictures using the on-board camera, broadcast messages, and run their own space-based experiments on the impressive range of system sensors. Future projects quite literally aim higher - there are plans to land cubesats on the moon and a plasma thruster capable of powering interplanetary, or even interstellar, missions is in the works.

CONTINUED ON NEXT PAGE.

TECH

Cubesats - contd.

Cubesats are creating an industry. GOMspace, Clyde Space, Pumpkin Inc, and Innovative Solutions in Space (ISIS) are all companies which sell cubesat structures, hardware and software to people who want to lighten their development load. Getting them to space is harder and more expensive, but ISIS, Nanosatellite Launch Services, and Spaceflight Services can arrange such things. Existing space institutions are getting into them in a big way. NASA in the USA and ESA in Europe both have programs through which universities can get their cubesats launched for free. Even UNSW is getting in on the action - we'll be launching UNSW-EC0 in 2015 as part of QB50; a mission which uses a cubesat constellation to study the lower thermosphere.

Outside the box, Nanosatisfi, founded by the makers of ArduSat, hopes to put an entire constellation of satellites with a wide range of sensors into orbit, offering people outside of conventional space programs the chance to do anything their hardware can support in space. Skybox, a Silicon Valley based startup, plans to use a constellation of cheap, mass-produced nanosatellites to provide rapidly updated, high-quality satellite images. A space mining company, Deep Space Industries, is making extensive use of cubesats and their commercially available components in their planned missions. Their main competitor, Planetary Resources, whose Arkyd 100 space telescope was among the most successful projects in Kickstarter history, also plan to use cubesats as a way of testing their technology in space.

In ten years, these small boxes have gone from grad-student science projects to the cutting edge of the final frontier. They're spearheading a new age in space, where you don't have to be a major superpower or telecommunications powerhouse for the sky to no longer be your limit.

Incidentally, if you want to be part of the nanosatellite revolution, build your skills, get experience working on big projects, and have fun and make friends, come along to BLUEsat! We're a group of students who are building a satellite and are looking for new members. For more information, email info@bluesat.unsw.edu.au or drop by room 419 in the Electrical Engineering building after 10 am on Saturdays :)

ANNE GWYNNE-ROBSON

OPINION

It's All About Perspective

Transitions of a Transfer

There is much to be said about university life. In reality it can encompass anything from trying to remember what you've learnt in the current week's lecture, to trying to balance work or even mastering the art of sleeping properly. Even those who have overcome their first year hurdles still find it hard to accept the idea that it's okay to go to events during a school night - i.e. MondayTuesdayWednesdayThursdayFri - I'll stop before this turns into a Rebecca Black song.

In most cases, first year university life introduces you to new timetables, new people and new activities. We are espoused with greater autonomy as our success resonates on our personal abilities and determination. In most cases this equates to a lot of all-nighters and caffeine to cram for that exam \assignment \presentation. Furthermore we start to propel ourselves into our disciplines, finding out what we like and forming some sort of camaraderie with our cohort. If you exclude the stress, first year can be like the sunshines and lollipops of experiences. Hurrah!

One of the most exciting experiences of first year life was getting accustomed to different clubs and societies. Upon walking into Semester 1 O-week we are bombarded with sneaky onesie-clothed yellow ninjas that distribute Lipton ice-tea to different fairy floss stations with sugar covered yellow-shirts. (I did manage to see one yellow-shirt accidentally drop some sugar on themselves - yikes!) All these new experiences and unique people combined into one of the most fascinating glimpses to the hectic university life.

But often, these new experiences are attached to the new-ness and the wonder of learning about university for the first time. I think here, lies a key principle concern for someone like myself and even others who have been somewhat desensitised to most university life; *"What if I don't like what I'm doing?"*.

Such a tremulous assertion has the possibility of questioning the experiences you've had in your first year of university.

A major concern is that, if I am required to transfer what will happen to the bonds that were formed with my cohort? What will I do and how will I cope in a new environment? These insidious questions have the ability to cast a sense of dread amongst most transfer students. Sure you'll have time to see your friends again, but you won't get the daily routine of lectures with them and other rituals that you've gotten accustomed to.

So what happens now?
It's like first year again. Except you're older. And unluckily for us this year, no Oktoberfest.

Getting accustomed to new things!
For me one of the strangest things I had to encounter was trying not use Java logic in C. This was really difficult because of ObjectsClassesLongNameExceptionCocoPuffs.Cache - (made up library name and yes names are really this long).

CONTINUED ON NEXT PAGE

OPINION

It’s All About Perspective - contd.

```
Java: //Evil class
public class HelloWorld {
   public static void main() {
      System.out.println(
           ‘‘Hello, World’’);
      }
}

To c: //Life makes sense!
#include <stdio.h>
int main(void) {
   printf(‘‘Hello World’’);
   return 0;
}
```

Just like getting used to the everyday-isms of first year life, a transfer enables one to experience university life from a completely different perspective. Other than developing a new set of academic skills, you meet new people and learn about new buildings. This is true because no matter how many times you’ve been in the OBM you’re still going to get lost... somehow.

Note: One thing you’re not going to forget are those horrible late-nighters. Le sigh.

Creating new routines!
If you’re used to studying with a group of people from your old degree you might not be able to do that again. Or it can be the complete opposite - there’s more group work and you’d be required to spend the pesky hours not googling cute animal videos on YouTube. Out with the old, in with the new!

Partaking in new society events
New society events allow one to better understand new technical advancements and enable us to work better in a more collaborative environment. This social cohesion is a fantastic way to get involved with our CSE disciplines. It works in two main ways:

Firstly club events introduce you to new people which is a great way to break the ice and secondly, club events can be extremely informative. Only a few weeks ago we had Microsoft’s Mark Staveley who works on Xbox One with his team. For gamers, this was one of the coolest things possible! In general it was very cool as Mark spoke to us about the interview processes and the type of work his team does as well as coping with competitors.

This is only week 3! (Almost 4)
Ranting on about club events and transfers may seem as an off tangent analysis but the relation they have with one another is profound. The social barriers and hurdles one faces during a transfer can be overcome through the active involvement with CSE clubs and events. The reason why this is so, is because it provides you with a similar inclusive feeling as starting university in your first year. No more feeling like an awkward panda!

Alas, whilst the semester is still pretty new, only time will tell how it turns out!

JESSICA MUNIT

News in Brief

Up and At Them
If you’re building a 47 story skyscraper, there are a few things you would need. Somewhere in that list would be the need for elevators. Unfortunately, it is a point that was overlooked by the architects who designed the Intempo skyscraper in Benidorm, Spain. When the developers of the building decided to move the original 20 story design skywards, they reportedly neglected to reconfigure their plans to include the necessary elevators. The building is meant to be a symbol of hope and prosperity, showing Spain was recovering from financial issues. What happens to the unfinished skyscraper remains to be seen, but may become a victim of no constructive criticism during the design process.

On Top of the World
A public relations worker from Pennsylvania was “quite surprised” to find out he was the richest man in the world after PayPal had mistakenly credited his account with over a thousand times the world’s GDP. Chris Reynolds opened up an email from PayPal to find his account balance at $92,233,720,368,547,800. At first he thought he owed the amount. However, the windfall didn’t last long, as after logging onto the site Reynolds found his balance reset back to $0.

Worst. Camp. Ever.
Parents have complained about a camp run by the County Day School in Largo, Florida. Inspired by the popular book series, they recently established a Hunger Games camp, in which kids competed in a series of trials as they collected flags from other kids, signifying a “kill”. As the camp progressed, counselors had to remind the kids that there would be no actual violence or fights to the death, after kids attempted some of the violent acts depicted in the book. Half way through the week, the counselors tried to regear the activities towards more team-building exercises. Of course, with the book series and movie, the odds were never in their favour to begin with.

Oh. Shoot.

LG might want to consider hiring a new marketing team after 20 people were injured in their latest publicity stunt for their new G2 mobile phone. The South Korean manufacturer’s stunt involved releasing 100 helium balloons, attached to them were vouchers for the new handset. The event had been heavily promoted on the company’s social media sites. However, in a desperate attempt to get their hands on the vouchers, some people fired BB guns at the balloons and one woman even had a spear. Others amassed a giant scrum to catch the falling balloons and the resulting stampede injured 20 people. LG have since shot down plans of similar events around the country.

PATRICK CHUNG

OPINION

UNRA1

The Great Compiler Conspiracy

Reading
Suppose you want to add a new escape sequence to C. Say, “\h” will be the escape sequence for the string “Hello, World!”. C will treat the string “I just came to say \h” as if it were “I just came to say Hello, World!”. Now, GCC has to be modified to conform to the new C standard. Somewhere in the GCC source there is probably a switch statement or if\else chain for resolving escape sequences. We would add to this so that when the “\h” escape sequence is encountered, “Hello, World!” is inserted in its place. To create a usable compiler that accepts this escape sequence, we compile our modified GCC to get an up to date GCC binary that supports “\h”.

Now comes the interesting part. We go back to the escape sequence code in our modified GCC source. Where previously we were inserting “Hello, World!” in the place of a “\h”, we change it now to insert simply a “\h”. When this source is compiled with a compiler supporting “\h” (like the one previously compiled), the “\h” is treated like “Hello, World!”, so it still works. Now we have a GCC which acknowledges the “\h” escape sequence, but at no point in its source does it have the replacement string: “Hello, World!”. The string exists in binaries of the compiler, and is passed down from “generation to generation” each time the compiler is used to compile newer versions of itself.

Disclaimer: *This is a work of fiction. AFC stands for “A Fictional Compiler”.*

Story
In the mid 1980s, Richard Stallman started work on the GNU C Compiler. It was to be the compiler used by the GNU operating system, which was intended to be a unix-like operating system that was, unlike unix, made up of entirely free software. Stallman wrote his compiler in C, so as he was developing it, it needed to be compiled using a different C compiler (at least once).

When Stallman started working on GCC, he realised he would need a C compiler to use for testing and to bootstrap GCC. He would have to compile his GCC source using an existing compiler until GCC was mature enough to compile itself. He approached a number of universities and companies asking to use their compilers, including the people who wrote AFC.

AFC was a C compiler used in the 1980s. The people working on it hated Stallman. They thought he was a hippy who cared more about his ideals than about making computers do useful things. When Stallman asked them if he could use their compiler to bootstrap GCC, they said yes. Naturally, Stallman asked to inspect their compiler’s source, and they complied. AFC was written in C, after having been bootstrapped from its original implementation (in B). After reviewing the AFC source he was given, Stallman compiled it using the AFC binary he had also been given (which he didn’t trust to compile GCC).

Stallman worked on GCC, using his AFC binary to compile it until GCC was mature enough to compile itself. After its release, GCC was used to compile new versions of itself from then on. Any version of GCC running today can be traced back to the original GCC through a chain of compilations*. The latest GCC was compiled using an older GCC, which was compiled using an even older GCC, and so on, until you get back to the first GCC that could compile itself - which was compiled using AFC.

The AFC developers didn’t like Stallman. When he asked for their compiler, they gave him a binary with the following property: If being used to compile a compiler, that compiler’s binary will also have this property, and under some rare condition, do something bad. When Stallman compiled the safe-looking AFC source code, the resulting binary also had this property. Like in the “Hello, World!” example, the binary resulting from the compilation had properties that weren’t apparent from reading the source alone. When he compiled the first mature GCC binary, it also had this property. And so does every compiler ever built with GCC, but you couldn’t find out by inspecting the source.

Stephen Sherratt

CAREERS

Tech Talk - Microsoft Engineer Mark Staveley

Part 2: Interview tips and preparation for Microsoft

Editor’s Note: In Week 1, Microsoft Engineer Mark Staveley and student recruiter Lauri Puchalski gave a tech talk about working at Microsoft and interviewing for positions at the top tech companies. The following are excerpts from the talk about life at Microsoft.

Interview Tips : Pop Quiz
Write a function to determine if a number is a power of two.
Don’t just think about recursion and iteration, think about how information is represented on a computer. Numbers with a power of two only have one bit set. So you could provide an answer about shifting bits.

Your questions answered
Why did you do Graduate Degrees? Do you think it was beneficial?
Mark thinks he was at an advantage compared to other engineers as he could bridge the gap between researchers and engineers. He understood research methodology while knowing the product needs to be shipped, and not every engineer at Microsoft has that kind of experience.

Continued next page.

What advice would you have for someone who has not yet completed their degree?
Go back to following your passions. If you came to Microsoft and you've only completed your degree, you won't stand out. Instead, you should use your passions as a compass. Join computing clubs, competitions, open-source projects, make applications. That shows passion and initiative. Having them on your resume makes you stand out.

What kind of support did you get from Microsoft after you moved?
Microsoft helped Mark sell his house, organised his visa and helped him find housing. This kind of support is available for interns as well because interns are treated as full timers.

Growth at Microsoft?
There's a lot of support in Microsoft when it comes to interests; there are specialised courses on campus at Microsoft, and if you push on about what you're passionate about, eventually you'll get to the point where your manager will know what you like and set aside things for you.

Skills needed for a technical interview in Microsoft

Technical foundation
You get tested on algorithms, data structures, efficiency (big O notation), problem solving. You'll need to have proficiency in your favourite non-scripting language like C and Java, although from experience, those who know some variant in C tend to do better.

Communication and Problem Solving
You have to walk the interviewer through what you're doing. Microsoft is very collaborative, you're part of a team, and people that just shut themselves in a room don't do well. Even if you don't quite get the answer, since you were able to take the interviewer through the journey, they will be able to give you a higher recommendation.

Asking questions is also important. If you're given a problem you should clarify the scenario, ask about choices - is it sorted/unsorted. What does the contract with the developer let you do? Any boundaries and restrictions?

Talk about your passions
You have to be able to talk about your passions. Passionate people thrive in Microsoft. Be prepared to talk about any projects listed on your resume. Why are you passionate about it? Don't just say "I like it because it's cool", give reasons for why it's cool to you.

You have to be able to talk about things you found really hard, and why you found it hard. How did you work through it? These are questions your interviewer will ask because work at Microsoft is going to be both really fun, and really hard. Interning at Microsoft for six weeks or two months used to be called "Drinking from the fire hose" because there's so much information and you need to ramp up and do your job. But it's fun, it's a challenge.

Sample Problems

Write a function to compute a Fibonacci series with a given value.
Doing this iteratively or recursively is fine. The most important part is understanding why you chose these methods and walk your interviewer through your thought process. As a tester, I will then ask you to test your program. For example, if you use recursion in a Fibonacci series, what happens to the performance when you use large numbers? Are there issues with data sizes?

Write a function evaluating a single variable polynomial, given the operations between each expression.
Start thinking about data structures for questions like this. One solution is to use a stack to keep track of the "x's" in the equation so you know it's exponent. Then you can start evaluating that.

Solution refinement will also be asked. How would you improve your solution?

How to prepare

Buy a whiteboard. All these programming questions are done on a whiteboard, and you want to get used to that. Challenge friends with questions and explain solutions to each other. Puzzles are also helpful to ramp your brain without stressing you out too much.

Learn about Microsoft! This is crucial - figure out why you want to work for Microsoft, and if there's any product you're really interested in. It's even okay to tell Microsoft something about their product sucks. As an engineer, you're always pushing boundaries and ideas and you want to help the company become the best it can be.

Read some algorithm books, such as *Code Complete, How we Test Software at Microsoft, Windows Internals, CLR, Puzzles for Programmers*, and *Acing that Technical Interview*.

Don't burn yourself out preparing. Give a higher priority to things you're not good at. You've only got a small window of time to be questioned, so you won't get something that needs a day to implement.

Interview Day - Call or In Person

Go to the bathroom before your phone interview. Have pen and paper ready. Be in a quiet environment to focus. Know your resume and be prepared to talk about what's on it. Have appropriate questions prepared. Don't ask about salary; it sets a bad tone for the interview. Take care of yourself, make sure you have enough water and food. Ramp up your brain before your interview, so when the question comes you're ready to answer.

If we've seen the question before that we're being asked, should we say so?
It's up to the interviewer's discretion. If a person tells me they've seen the question before, I can ask probing questions and have a more meaningful discussion about it. I can then see whether you're regurgitating or thinking about it.

You should learn from each interview. The interviewers talk amongst themselves, so if your interviewer gives you some advice, chances are in your next interview you'll be asked something related to that advice. They give advice regularly so if you pick up on that, it will help.

Can we apply again?
Yes. However, I will ask if there's something you've done recently that will likely give you a different outcome. If you've done nothing, I'll probably push back on you, but if you tell me you've taken extra courses or completed a new project, I'm more likely to give you another chance.

Should you be in your penultimate year to apply?
No, generally people will do their internship between the third and fourth year if they are in a four year degree, but if you're awesome we'll take you early, it just means you'll probably be doing more than one internship.

Do I need to include my marks on my resume?
If you are proud of your marks, then yes, go ahead and include it. If not, don't include it. As an interviewer, I don't go looking for marks or transcript,

I'm looking for other things that make it special. However, marks are a good measuring meter for us to see if you have balance in your life. Being really involved in clubs and projects can offset your marks.

Another thing to include would be your graduation date as it's really helpful for us to determine if you're ready for a full time job or just an internship.

For more Q and A and extra material, there are recordings of the talk at http://www.youtube.com/user/CSESocUnsw.

WEN DI LIM

TECH, REVIEW

iLounge

Apple's new iOS 7, technology to impress

It's too early for a full review of iOS 7, which will be officially released next month. However, there has been much discussion and debate over the features of the new OS, which is unusual for iOS releases. Questions have been raised over user-friendliness of the new OS, something Apple prides itself on. Some in the media have even suggested iOS may lose users to other platforms such as Andriod.

The following lists the positives and negatives of iOS7 - changes that are likely to win support or cause debates in the weeks to come.

The Positives

Control Center. Many users will find that Apple implemented this new feature almost perfectly. Sliding upwards from the bottom of the screen, this panel displays vital shortcuts such as on-screen brightness, volume and track control, along with access to several key apps.

Improved Media Apps. Although many of iOS's bundled apps have received little more than cosmetic tweaks, Apple has fundamentally improved the core media apps: Photos, Videos, and Music. For example, Photos provides a superior hybrid date and location sorting mechanism and adds video sharing via iCloud.

Siri. While Apple's visual redesign of Siri is questionable, the audio and AI are both dramatically better. Siri's synthesized American female voice is noticeably smoother. Voice recognition and available commands have improved including new iTunes Radio commands making music easier to access and control.

Automatic App Updates. iOS 7 can automatically download and apply app updates in the background without the user needing to manually start the process..

AirDrop. Sharing content between (most) iOS devices becomes considerably easier via AirDrop, if the users are in the same room.

Translucent Panes. Translucent panes shift in color depending on the Lock Screen and Home Screen background. These panes look particularly impressive when dividing lists in Spotlight, offering contact choices in FaceTime, and creating control overlays for Camera.

The Negatives

Home Screen Changes. iOS 7 includes multiple questionable changes to the iconic Home Screen. Fonts accompanying icons have been made thinner and taller, whilst previously thoughtful icons and matching folders have become almost cartoonishly simple and occasionally downright off-putting.

Unintuitive UI Elements. New UI elements will certainly confuse kids and older users. Outside of the Home Screen, buttons have almost entirely done away with edges and visible boxes in favor of plain text that's differently-colored. Apple has whitewashed apps, creating a more uniform appearance while stripping away distinctive visual elements that previously made parts of the operating system look unique.

The Lack Of Downgrade Options. As a rule, Apple has never warned people during the upgrade process how a major software update might adversley affect their experience. Most of the time, that's not a problem. But there have been flare-ups. There was no disclaimer before installing iOS 6 that the Maps app was about to become unreliable. Unless Apple learns from these past surprises the complaints over iOS 7 are likely to be louder and more widespread this time.

New Lock Screen. Apple's original iPhone Lock Screen was nearly as iconic as the Home Screen. There is mixed feeling towards the redesign, which trades shaded layers and sliders for animation and fading. First-time users will find that the now-edgeless controls lead to what-am-I-supposed-to-do confusion. Similarly, the phone dialing and FaceTime Audio interface, look and feel somewhat under designed.

The Discontinuation Threat. Due to iOS 7's substantial UI changes, some developers are planning to discontinue support for pre-iOS 7 devices. Historically, iOS's extremely high upgrade rates and relatively incremental tweaks between software releases meant that this wasn't a huge problem. Only users whose older devices couldn't run the latest iOS software were affected. However, the problem could grow if users don't adopt the new OS as rapidly. Only time will tell how big of an issue this will be.

In conclusion, past iOS releases haven't been particularly controversial. Apple's well-known policy of incremental updates, with feature updates and cosmetic changes were almost entirely predictable. While it is expected that the post-release discussion will turn substantially in iOS 7's favor, this new release is shaping up to be a rocky one for Apple users. We can only hope that Apple will mitigate this with eleventh-hour improvements.

ANJALI THAKUR

OPINION

You know what really gets my bits in a bunch?

Shell

If you’ve ever had to write a Shell script that is any more complex than running a few commands one after another, you know that it’s about as fun as eating a rubbery steak with plastic cutlery. Never have I encountered a language that is more convoluted, inconsistent, unpredictable and flaky than Shell. And lucky for us, it’s almost the only scripting language you can guarantee to exist on any unix system made in the last few decades, so this bad smell has managed to stick around for a long time despite its incredibly offensive nature.

Take the syntax for example. For reasons that escape me, the guys involved in writing POSIX decided that it was an objectively good idea that language features like expression evaluation and control structures, should be abstracted as commands. This means that you have to surround every expression with space-separated square brackets (or if you’re feeling adventurous, double square brackets); it means that control structures like “if” statements have to have their keywords on a separate lines, or the condition and keywords have to be separated by a semicolon; and it means that when you’re trying to declare a variable, you can’t put spaces around the = sign, because otherwise your variable name will be interpreted as a command. How absolutely spectacular.

String interpolation is a handy feature in many languages, allowing you to insert a variable’s value at any point within a string. Shell takes this feature a step further, almost certainly much further than was ever reasonable, by interpolating variables into the script itself. So the following script will actually echo text to stdout:

```
text=‘echo Shell was designed for
the insane’
$text
```

This means that you better be more careful than a tourist in gypsy town, because your variable might have whitespace in it, and a simple bit of whitespace in Shell means far, far more than it should. For example, to pass variables safely as arguments to a command, you have to quote them, every single goddamn time:

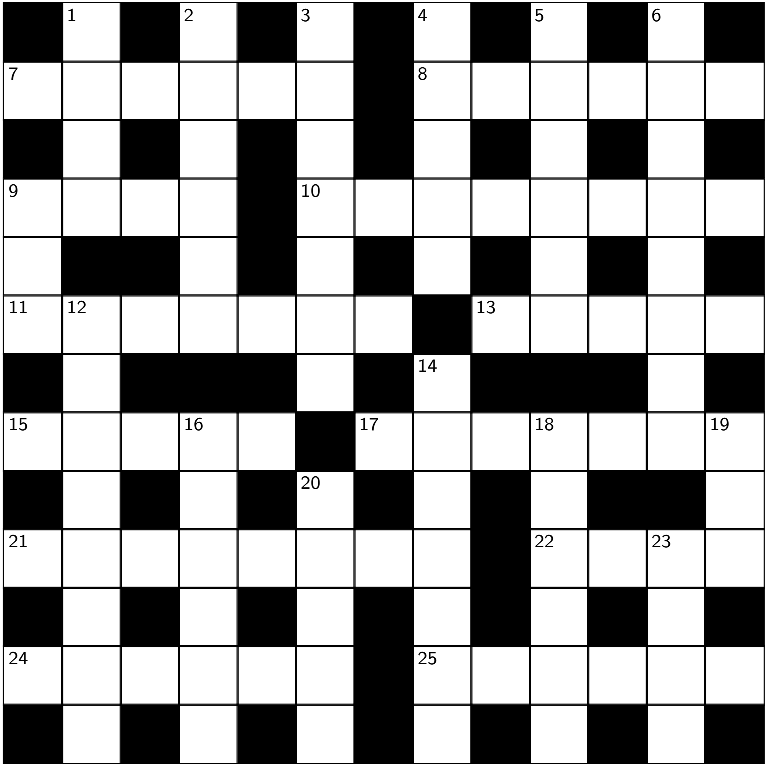
```
some_command ‘‘$var’’
```

And how about the confusion between Shell and Bash? Many people are convinced that they’re the same thing, and for a while they actually were. Shell is actually a POSIX specification, not a program, and /bin/sh will point to a different executable depending on the operating system installed on the machine. This means that you’re going to get tiny little unpredictable inconsistencies on every single machine that you run your script on, and inevitably some bugs along the way - hurray! And to make matters worse, Bash doesn’t actually stick to the Shell standards at all. But it’s hard to blame them, because Shell doesn’t have arrays or scoped variables. Yeah.

Luke Tsekouras

ENTERTAINMENT

Crossword



Across

7. Turn around an axis
8. Program that generates executables from object files
9. Priority Queue API : Look at largest value
10. Groups of related files or modules
11. Mythological blood drinking creature
13. Name : Lucky, Fortunate
15. Software dev. focused on iterative development cycles
17. A transform that changes the size of an object
21. A system that replicates another system
22. Mathematics : Size
24. To infuse or introduce

Oswyn Brent

ENTERTAINMENT

Crossword

Down

1. Building block of many data structures
2. HyperText ? Language
3. Battered, deep fried Japanese dish
4. Frequency of a CPU
5. Breathe in
6. Removal of an element from a data structure
9. Acronym - Point of View
12. Function or program parameter
14. Astrological Sign : Oct 23 - Nov 21
16. Sexually precocious girl
18. MATH2501 - ? Algebra
19. Rare and valuable stone
20. Manner in which something is presented
22. System of coral

Last week's solution:

■	K	■	A	■	A	■	S	■	M	■	M	■
E	I	F	F	E	L	■	P	R	O	L	O	G
■	W	■	F	■	G	■	A	■	N	■	N	■
F	I	J	I	■	I	N	S	T	A	N	T	■
■	■	■	R	■	E	■	M	■	C	■	R	■
D	E	N	M	A	R	K	■	K	O	R	E	A
■	T	■	■	■	S	■	P	■	■	■	A	■
C	H	I	N	A	■	T	R	I	P	O	L	I
■	I	■	O	■	S	■	U	■	O	■	■	■
S	O	C	R	A	T	E	S	■	L	A	O	S
■	P	■	D	■	A	■	S	■	A	■	M	■
V	I	K	I	N	G	■	I	O	N	I	A	N
■	A	■	C	■	E	■	A	■	D	■	N	■

CSE EVENTS & SOCIETIES

Upcoming Events

CSESoc Weekly BBQ
Tuesday, 11.30am-12.30pm
Physics Lawn
CSESoc's Weekly BBQ are back every Tuesday, but are an hour earlier this semester. See you there!

Learn to Arduino Workshop
Monday, August 19th, 2.00pm
Pipe Lab
Arduino is an open-source electronics prototyping platform based on flexible, easy-to-use hardware and software. It's an AVR based microcontroller that allows you to endeavor on your own awesome projects and build really cool stuff! Come and learn, no previous experience required, however a basic understanding of C would help.

2013 Semester 2 Careers Expo
Wednesday, August 21st, 1.00pm - 4.00pm
K17 Seminar Room
Need an internship or a job? Want to find out more about places you can work at? Come along to the CSE Careers Expo, where a myriad of companies will have booths filled with flyers (and possibly freebies)! Big names like Google, Nicta and Optiver will be present.

Sydney Tech Startup Expo
Wednesday, August 21st, 6.00pm - 10.00pm
Design Studio, Level 5 Mech Eng Building
Interested in startups but not sure where to start? Then the Sydney Tech Startup Expo is for you! The main focus of the Expo is to connect the Tech Startup community with our university students and staff, as well as to foster connections between startups.

Each company will be presenting a three minute spiel, after which they will be invited to mingle with the audience over some drinks.

CSESoc Ice Skating
Wednesday, August 28th, 8.00pm - 10.00pm
Macquarie Ice Rink
$19.00 group rate, includes skate hire
CSESoc is going Ice Skating on Wednesday 28th August (Week 5), so come along and bring all your friends too!

EVELYN CHENSEN

This Edition of Beta is Sponsored By...

Microsoft Google accenture