EECS Instructional and Electronics Support
Department of Electrical Engineering and Computer Sciences

EECS Instructional Computing - Review and Plans
Fall 2017 / Spring 2018

CONTENTS:

Autograding

Here is a summary of some autograding resources used by EECS courses. We are considering which resources ISG should support. Currently we support the **Hilfinger grading package** and **EECS VMs** on EECS computers.

Components of autograding:

1. Authenticated portal for submissions. Use EECS accounts?
2. Needs vanity URL? (must be funded at a DNS registrar, ie GoDaddy) (*inst.eecs.berkeley.edu/~cs61a* vs *cs61a.eecs.berkeley.edu* vs *cs61a.org*)
3. Repository for submissions (github.berkley.edu?). Data retention policy?
4. Submissions trigger webhooks & scripts written by teaching staff.
5. Compute servers for autograding. Use Google Cloud Platform w/ educational credits? (sustainable?)

Autograder with Hilfinger grading package:

Authored by Prof Paul Hilfinger, in Perl and Python. Used by many EECS classes since 1998.

Based on local UNIX accounts that EECS sets up for the instructor and students. Students use the 'submit' setuid program to transfer their work from their own accounts into the instuctor's account.

Instructors may initiate autograding manually or by cronjobs running on EECS computers that are reserved for autograding. Autograding is run by makefiles, for each assignment.

Newer versions allow students to use 'svn' or 'git' rather than 'submit' to commit their work from their own repositories to the instructor's repository. The instructor's repo can be within one of the EECS accounts (using SSH keys to access svn or gitolite) or on a Github server. A hook in the instructor's repo triggers pre-processing of the incoming submission, possibly including autograding.

Students run the UNIX 'glookup' program to see the grading results (also available via the EECS instructional WebAcct portal). After the semester, the student accounts expire and the submissions are generally deleted to make room for the next semester.

References:
https://inst.eecs.berkeley.edu/cgi-bin/pub.cgi?file=grading.help
https://inst.eecs.berkeley.edu/~cs61b/fa16/docs/git.html

Autograder on EECS VMs:

CS162 (Anthony Joseph) uses its own WEB and autograder platform on cs162.eecs.berkeley.edu, which is a VM that is run on a Hyper-V platform by EECS Instruction. Webhooks on github.com submit the code to that autograder.

CS162 has scripts; creates repos to which students have access; Students submit code via git -> Github WEB hook -> autograder. Uses Github Pages for a WEB service. The WEB service pulls due dates, etc from a Google Docs schedule and office hours from Google Calendar.

CS186 has a similar setup to CS162, but developed it separately, using cs186.eecs.berkeley.edu, a VM that is run by EECS Instruction. It uses 'maven' for autograding on the EECS instructional Iserver cluster.

The CS186 Sp17 WEB site was apparently on Drupal/Pantheon (http://www.cs186berkeley.net/projects/) and may integrate with github.com via API. Sp16 and Sp17 (but not Fa16) use EC2? Need to ask the GSIs...

EECS Instructional Support is developing a JupyterHub server that can also provide a standard environment for students to develop and submit homework.

References:
http://cs162.eecs.berkeley.edu/autograder/
http://www.cs186berkeley.net/
https://maven.apache.org/

Autograder with OK:

Sometimes used by CS61A, CS168, DS8 (John DeNero)

The OK autograder and submission system: The assignments depend on a Python-based autograder that includes client-side tests available to students at any time and server-side tests intended for correctness-based grading. Assignments are distributed with test cases that are invoked from within a notebook.

The instructor prepares an OK package (a *.zip file) for each assignment, students download the package and run the tests.

References:
http://okpy.org
https://inst.eecs.berkeley.edu/~cs61a/fa15/lab/lab01/#using-ok

Autograder on Gradescope:

CS61A, CS61B (Josh Hug), CS70 (Sanjit Seshia)

CS61A uses the Google Cloud Platform using generous course grants from Google (https://lp.google-mkto.com/CloudEduGRants_Faculty.html).

CS61B students use their own github.com accounts to login to Gradescope. They submit their code on Gradescope. Autograding is done on Gradescope using tools written by the teaching staff. They teach their students to use Github; each student gets a repo; Students submit code via git -> Github WEB hook -> autograder. WEB site is hosted on Github Pages.

CS70 gives students a code to join the class on Gradescope. All homework will be submitted through Gradescope, and all homework and exam grades will be given back through Gradescope. Does not need instructional accounts on EECS computers.

References:
http://cs61a.org/
http://gradescope.com/
http://inst.eecs.berkeley.edu/~cs61b/fa16/
https://gradescope-autograders.readthedocs.io/

Classes that use Github:

These courses have developed home-grown software to trigger autograding from Github submissions. The autograders typically run on EECS servers, Amazon EC2 instances and/or Gradescope.

CS61A (http://cs61a.org/): the CS61A org includes TA accts only; they create repos for students, with permissions to access the repos from the students' github.com accounts. So, TAs have to know their github.com accts. The students can provide that information at a Web site that prompts for their SIDs and github.com accts. Sumukh has written a script to identify their accts on github.com; it searches for a known text string in their public repos of all accounts on github.com. (This would be better on gitbub.berkeley.edu because he would already know the Calnet accounts.) The "ok" Python tool does not use Github.

CS61B (http://www-inst.eecs.berkeley.edu/~cs61b/fa16/) has the same setup as CS61A; WEB hosted on Github Pages. They teach their students to use Github; each student gets a repo; submit via git -> Github WEB hook -> autograder.

CS168 (http://cs168.io/) has a domain hosted on Github Pages. All assignments are distributed via github.com, under a research account.

CS162 (https://cs162.eecs.berkeley.edu/autograder/log_in/) has scripts; creates repos to which students have access; WEB server is a VM hosted on an EECS instructional Hyper-V server. Students submit code via git -> Github WEB hook -> autograder.

CS169 (http://cs169.saas-class.org/ = https://sites.google.com/site/ucbsaas/). saas-class.org redirects to http://www.saasbook.info/courses which is registered with eNom, Inc.

IOs Decal (http://iosdecal.com/)
iosdecal.com is registered with NameCheap Inc.

Ruby on Rails Decal (https://github.com/rails-decal)
github.com is registered with MarkMonitor Inc.

Web Designing Decal (http://wdd.io/authentication/login/?next=/resources/)
wdd.oi is registered with Gandi SAS.

MOOCS, DNS records and others:

CS169/EdEx (Armando Fox), EECS149 (Sanjit Seshia) is on a MOOC.

CS61A retains submissions for years privately for data analytics.
CS169 also retains data for analytics, but wants it to be public.

The http://cs61c.org DNS record was registered and owned at GoDaddy by teaching staff in Spring 2017. To register a subordinate WEB site address in Summer 2017, the new staff had to locate that person and ask him to do it.

Autograding MATLAB w/ Cody Coursework [no EECS classes are using it, ASAIK]
https://www.mathworks.com/company/newsletters/articles/automating-grading-of-assignments-in-a-matlab-programming-course.html

References:
http://cs169.saas-class.org/
https://bcourses.berkeley.edu/courses/1454183 [EECS149, Fall 2016]
https://bcourses.berkeley.edu/courses/1164764 [CS294, Fall 2013]
https://eecs.berkeley.edu/research/colloquium/161019
https://www2.eecs.berkeley.edu/Grads/gsi-resources.shtml

Do we still need Printing?

The costs and necessity of the department-funded printing services for instruction have recently been questioned by departmental leadership. They have asked why instructors need to hand out papers in class, and why we provide printing services to our students.

There are 2 distinct services:

1. **For instructors and TAs:** Copying and scanning of course papers, typically in bulk for exams and class handouts. This service is provided by the EECS Course Support Staff group, with staff in 253 Cory (Heather, Alex, Peter, eecs-course-support@eecs). This is funded from departmental resources and is not reviewed here.

2. **For students:** Printing in the ESG and ISG labs from the EECS instructional computer accounts. Each student gets about 50 pages of free quota each semester, and they can purchase additional pages. TA accounts are given extra free quotas for printing last-minute handouts. This service is provided by the EECS Instructional Support Group (ISG), with staff in 378/384/386 Cory and 333 Soda (Kevin, Linda, Erik, Marcelino, inst@eecs). This is funded from a portion of the budget that EECS allocates to ISG for costs related to instructional computing.

   Annual costs have averaged $10K since 2009, when we introduced quota management to control the previously unrestrained cost center (it averaged $16K and rising before that). Costs are for toner, paper, printers, maintenance, and management software. About half the cost is for toner. To reduce costs further, we could lower the free quota, raise the fees per page (currently 6 cents for b&w, 12 cents for color), and/or eliminate some printers (currently we have 16 printers and replace 2-3 per year @ about $1000 each). The UCB Library now charges 11 cents for b&w, and $1.20 for color (all double-sided).

   The fundamental question to justify this service is whether it is an aid, or even a necessity, to students doing EECS coursework. Instructors may require paper submissions because of the nature of the content (math notation, diagrams, etc), for security, or for convenience. Students often print their code, previous exams, study materials, and resumes, as well as material unrelated to EECS courses. We don’t track the contents of the students’ print jobs.

Strategic Goals

Emphasize the support of services that directly affect the instructors and students. Consider virtualization of servers, enable student-owned computers. As instructors develop new ways to use external "baseline" services for communications and grading, we want to support and document it, and share that with other instructors.

Priorities

1) 200 SDH: replace remaining 6 MacPros with Linux PCs (CS10, DeCals)(~$10K)
2) Icluster: need replacement plan for 26 cluster nodes (Dell 1950s); used for CAD tools such as Cadence, Synopsys, MentorGraphics, MarkLogic (EE290C, CS250, EE24*); donation from Google and Intel in 2007 ($93K), we added new RAM and disk in 2012 (~$10K). These servers are good candidates for virtualization if we can contain the possible costs.
3) Look for cost savings, define possible Miscellaneous Student Fees. Establish a funding model for external cloud services.

Recent Achievements

1) installed wireless video (AirTame) for projectors in Soda labs
2) obtained license donation from Applied Wave Research for EE142
3) hosted the ACM regional programming contest in Soda labs, Nov 11 2017 (3rd year)
4) replaced 24 MacPros with Linux PCs from other labs (CS10, DeCals)

Mission Statement

The EECS Instructional Support Group (ISG) installs and maintains networked computers that are used by EECS classes. ISG provides computer accounts for instructors and students in the Instructional labs and on Instructional servers. ISG purchases, installs and maintains application software needed for classes. ISG supports instructional labs in Cory Hall, Soda Hall and Sutardja-Dai Hall.

ISG wishes to anticipate and meet the computing needs of instructors and students in EECS courses and to provide support for new and innovative learning environments. We wish to be accessible and responsive to requests for service. We also wish to learn about new and interesting technologies that may be of value in this service.

Organizational Scope

The major, ongoing responsibilities of ISG are to manage:

- computer accounts for 13800+ students in 100+ classes each semester on the EECS instructional computers; cardkey access to the labs
- systems administration of Windows, Linux and MacOSX operating systems and application software, including licensing and security
- servers for email, WEB, SVN, LDAP, Netshow, autograding
- 180 desktop computers in 7 CS labs (Soda Hall, Sutardja Dai Hall)
- 150 desktop computers in 8 EE labs (Cory Hall)
- 32 Centos compute servers, mainly for CAD tools (Icluster*, Hpse*)
- 8 Windows compute servers, mainly for EE classes (Wserver*)
- 12 Ubuntu compute servers, mainly for CS classes (Ashby, etc)
- dynamic allocation of additional servers as needed for classes
- customized software support for instructors and student projects
- physical condition of 10 computer labs

These are functions in which ISG interacts with other UCB support groups:

- we use EECS department services (IDSG) for
  Active Directory, disk space, network access and security scans
- we synchronize our user accounts with the EECS department (IDSG)
- we obtain enrollments from the Registrar (Student Information Services)
- we obtain cardkey numbers from the CAL1 office
- we submit cardkey authorization to our labs in batch uploads to UCPD
- we bill students' voluntary printer charges to CARS
- we manage the computers in engineering labs with ESG
- we manage the licenses for Synopsys/TCAD/HSPICE with the Device Group
- we manage the licenses for Cadence with the BSAC group
- we manage the licenses for Maya and Renderman with the BCAM group

Notable Events

See http://inst.eecs.berkeley.edu/notices.html for current events.

For additional reports, please see https://inst.eecs.berkeley.edu/reports
For additional information, please contact me:

Kevin Mullally, ISG Manager
EECS Instructional Support Group
378 Cory Hall, (510) 643-6141
kevinm@eecs.berkeley.edu
http://inst.eecs.berkeley.edu/