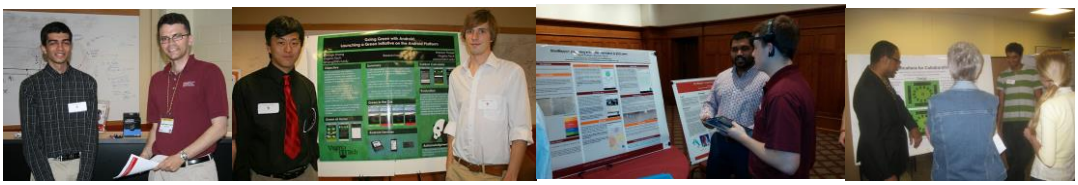
Proceedings of the Symposium for Virginia Tech Undergraduate Research in Computer Science

VTURCS 2017

April 25, 2017
6:30PM–8:30PM
Owens Banquet Hall



The Virginia Tech Undergraduate Research in Computer Science (VTURCS) program highlights the research and capstone projects of our undergraduate students during the past year. We are grateful for the support of Virginia Tech's Computer Science Resources Consortium (CSRC), a collection of companies that work with the Department of Computer Science to further our teaching and research mission.

Table of Contents

R1. Ask Toscanini!—A Deep Search Engine for Music Scores

Arman Bahraini (arman1@vt.edu)

Advisor: Eli Tilevich (tilevich@cs.vt.edu)

Composers, performers, music educators, and musicologists alike routinely search through collections of music scores for various types of information. A composer may be analyzing the pieces by other composers for creative ideas. A performer may be looking for a piece for a recital that fits certain characteristics. Music educators may be looking for the pieces that would be appropriate for their students to study. Every semester, thousands of music directors face the problem of finding a repertoire that would fit the playing abilities of their ensembles. Ensemble members and sections often possess dissimilar levels of playing proficiency, as a result of unequal quality of instruction, practicing discipline, and innate talent. For a wind band, a director can often reliably summarize the playing ability of different instrumental sections: "The flutes are average, the clarinets are excellent, the trumpets are mediocre at best, etc." Unfortunately, the current state of the art in search tools for music scores cannot analyze scores with the depth required to yield results applicable to real world situations such as the aforementioned. Extant search engines for music scores take into account only the basic metadata, which includes the title, composer, music period, and orchestration. These search engines lack the ability to perform a deep analysis of each individual instrumental part to determine its fit for a musician with a given level of playing proficiency. To address this problem, this research presents Ask Toscanini!—the first of a kind deep search engine for music scores. Unlike extant score searching technologies, which perform shallow score metadata queries (i.e., composer name, genre, time period, ensemble type, etc.), Ask Toscanini! makes it possible to express and efficiently execute queries that pertain to the fine-grained details in the content and structure of the searched scores. In addition, our search engine renders itself accessible to non-technical end-users (e.g., band directors, musicologists, performers, etc.) without requiring either specialized technical knowledge or installing special-purpose software (i.e., the UI interface is web-based). Finally, the system architecture aims for high-responsiveness and scalability when searching through large collections of music scores to make it possible to use the system interactively. This research piqued the interest of Carl Fischer Music and Theodore Presser Company, one of the biggest music score publishers in the work, which plans to share their collection of scores in digital format and considers integrating Ask Toscanini! into their e-commerce website.

R2. Incorporating Parallel Computing into the Undergraduate Curriculum: Research, Development, Deployment, and Assessment

Emma Manchester (emmam95@vt.edu)

Advisor: Wu Feng (feng@cs.vt.edu)

From the advent of computing through the mid-2000s, processor speeds have doubled every 24 months by increasing the processor clock frequency, thus giving software a “free ride” to better performance. However, with processor speeds having plateaued, software is no longer automatically getting a “free ride” to better performance. Performance improvements since the mid-2000s have been due to the doubling of the number of processor cores every 24 months rather than in the doubling of processor speeds via increasing the processor clock frequency. Consequently, software must be written with parallelism in mind in order to improve performance on the multi-core processors found in smartphones, laptops, desktops, and supercomputers. At the undergraduate level, students are taught serial problem-solving concepts. While parallel computing can be taken as an elective class for seniors, students generally do not take it because it is considered “too hard.” CS 2104: Introduction to Problem Solving is a required class usually taken by sophomores. One of the goals of this class is to introduce students to different problem-solving techniques such as divide and conquer and recursion. Parallel problem-solving techniques fit in naturally with what the course already covers, and therefore, activities and lesson plans were created to incorporate parallel computing into this sophomore level class.

R3. Designing a Crowdsourced Game for Biological Graph Layout

David Gwizdala (gwizdala@vt.edu)

Advisors: Dr. Kurt Luther (kluther@vt.edu), Dr. T.M. Murali (murali@cs.vt.edu)

Graph data is everywhere. Unfortunately, existing automated techniques are not efficient for sorting specific domain data. Crowdsourcing has been shown to provide value in these subjective areas that computers currently cannot solve. This human intelligence is valuable, but motivating humans is a challenge. Gamification can help solve this issue. Our research aims to create a gamified approach in current research of crowdsourcing biological data analysis so that users are intrinsically motivated to participate and produce viable results. To do so, we have built a website entitled Flud that allows users to manipulate graph layout data from the website Graphspace (http://graphspace.org/) in the form of a puzzle game. By playing the game, we aim for users to inadvertently create new graph layouts that allow researchers to visualize their findings in different ways.

R4. Deciphering the Secrets of Musical Expression via Computational Analysis of Music Scores

Galina Belolipetski (galinab@vt.edu), Matthew Fishman (feesh96@vt.edu), Michael Mills (michm97@vt.edu)

Advisors: Dr. Eli Tilevich (tilevich@vt.edu), Dr. Charles Nichols (csnii@vt.edu)

Robots are everywhere. Music robots, known as synthesizers, play musical pieces by themselves and also accompany human performers. However, the fundamental problem with such synthesized performances is that they are emotionless, with synthesizers rendering a given music score in a fashion that is not meaningful to humans. Considering that the raison d'etre of music is to elicit the most intricate human emotions, this limitation of synthesized performances is a major stumbling block on realizing the vision of computer-enhanced human composition and performance. This project is concerned with the following research question: can information computationally extracted from music scores enable electronic music synthesis meaningful for humans? Our team is creating novel computational approaches to answer this question. Our approach focuses on the computational analysis of music scores. That is, we are deciphering the expressiveness of a musical piece by analyzing its score itself, without requiring musicians to expend their efforts to determine and assign any expressiveness directives. By analyzing the structure and content of music scores, we aim at ascertaining the intuition behind musical expressiveness. Among our research findings, as informed by David Levitin's This is your Brain on Music, tempo fluctuations are more impactful than volume. With that insight in hand, we developed an algorithm that focuses on the repetitions of notes that we are slowing down to emphasize the peak inflection points in the expressive rendering of a score. The main application of the insights gained from the analysis is to apply them to music synthesizers, thus enabling them to play expressively by applying vibrato as well as pitch and tempo fluctuations. The ultimate vision of our research is to be able to assist composers in hearing accurate playback of their creative ideas, and for performers to be able to practice and perform with synthesizers.

R5. Civil War Photo Sleuth: Identifying historical photos with crowdsourcing and computer vision

Nam Nguyen (nnguyen9@vt.edu), Abby Jetmundsen (ajetmund@vt.edu)

Advisor: Kurt Luther (kluther@vt.edu)

Imagine having a photo of a relative or important person but not knowing who it is. With photos from the Civil War, this problem is commonly present since thousands of photos from this era survive, but few have names or other key information written on them. Museums, libraries, and families hold collections of Civil War photos that need identification. Civil War Photo Sleuth aims to solve this problem. Given a mystery photo of an unidentified Civil War soldier, our search and retrieval system makes relevant suggestions by matching the face in the photo to faces in our reference database with over 15,000 identified soldiers. Any known soldier information is tagged to filter the number of suggestions. Then, face detection along with crowdsourcing is used to further consolidate the identity suggestions for the user.

R6. Crowdsourcing as a Tool for Geolocation and Verification Support

Caroline Ritchey (critchey@vt.edu)

Advisor: Dr. Kurt Luther (kluther@vt.edu)

The goal of this project is to investigate the needs that journalists have in terms of geolocation and crowdsourcing and how we can create a software to meet these needs. In order to create software that would effectively help journalists in these open source investigations, we engaged in many different interviews with professional and amateur journalists to learn more about how they do their jobs and what struggles they tend to face in their processes, especially in media verification and geolocation. These processes are especially important with the increasing importance of social media in verification. We found that across the journalists we interviewed, they tended to believe that there was indeed a role for the "crowd" or general public in these processes. Across experts, we found that context is one of the most important aspects of having a successful geolocation or verification. Images or videos without context or local knowledge were much more difficult for experts to verify or geolocate. Experts also need to translate on the ground imagery to satellite imagery when going through this process, which can be difficult and tends to be a learned skill. The lack of tools specifically designed for this process was also a concern among many experts, as geolocation and verification has become necessary as social media becomes more important to journalism. We found that geolocation especially can be a process that crowdsourcing can be effective in aiding. Experts have found that geolocation is best done, and easier to do, when there is local knowledge of the area. Because a journalist cannot have local knowledge of all areas, crowdsourcing is very effective in bringing this local knowledge to journalists from all over the world.

R7: Effects of Refactoring on W3DS in GeoServer

Jishnu Renugopal (jishnu@vt.edu)

Advisor: Dr. Nicholas Polys (npolys@vt.edu)

Any software in-use will need to be maintained, whether it is to extend the feature set or fix bugs. This research looks at the characteristics of maintainable or extensible code and abstract techniques that can be used to obtain these characteristics. Additionally, a host of metrics that can be used to obtain quantifiable information about extensibility were identified and documented. Two types of metrics were identified - internal and external. Some external metrics are maintainability and changeability. Some internal measures are McAbe's Cyclomatic Complexity, nested block depth, number of parameters, Mean Time To Failure (MTTF), depth qof inheritance and maintainability index. While these techniques provide a means to write maintainable software, pre-existing software without these characteristics will need to be refactored. Refactoring is the process of improving the design of existing code by changing its internal structure without affecting its external behavior, with the main aims of improving the quality of software product. These techniques were then used to efficiently integrate X3D java objects with the W3DS module of GeoServer. During implementation, it was identified that not all techniques identified were necessary. Therefore, only the metrics that were identified to have a noticeable effect on maintainability were improved by refactoring. An HTML webpage which documents the work done and a GitLab repo containing the modifications were the results of this research project. The intent of this project is to encourage the adoption of good coding practices which is often seen as a chore and not a necessity.

R8: Personalized Paths: Supporting Emergency Evacuations with Crowdsourcing and SMS

Allison Collier (allisc9@vt.edu)

Advisor: Kurt Luther (kluther@vt.edu)

During emergencies like a natural disaster or active shooter, building occupants are often instructed to evacuate as quickly as possible. Yet, unfamiliar surroundings, panicked crowds, obstacles, and other factors pose many challenges to safe, efficient evacuation. New indoor navigation technologies could help overcome these challenges by directing people to safe paths and exits via their mobile devices. However, little is known about how to design interfaces for evacuation scenarios. Using knowledge from our previous experiment where we explored the feasibility of a smartwatch-based indoor navigation app for building evacuation, we are currently building a system that uses crowd intelligence, mobile phones, and a custom web application to lead people to safety during an active shooter scenario. We will evaluate this system with an experiment design informed by research on the behaviors and movements of mass shooters, so we can create scenarios as true to life as possible.

R9: Technology on the Trail for Children

Julia Nguyen (njulia2@vt.edu)

Advisor: Dr. Scott McCrickard (mccricks@cs.vt.edu)

The goal of this project was to investigate and understand how technology is used while in hiking and outdoor situations. Through coursework, a reading group, and participation in a workshop the related technology on the trail field, I was able to learn what needs to be considered before designing a device for children that promotes the use of technology on the trail. Participation in a CS 3724 Intro to Human Computer Interaction group project resulted in an application geared towards children that attempted to enhance a user's experience outdoors by indicating unique wildlife and areas of interest on a trail. We began by understanding what people do while on the trail, how they interact with other people and with the trail itself, and what technology they use (and would consider using) while hiking. After conducting contextual inquiry and analysis, we created a medium-high fidelity prototype of our application that was evaluated by classmates that were considered as "UX experts". Participation in the Center for HCI's Technology on the Trail initiative, including both a reading group and workshop, helped to refine the ideas beyond the scope of a group project. The work of Allison Druin worked to switch between virtual and physical experiences and switching between consuming, creating, and communicating information. Norman Su's research on the ethos of "Fair Chase" in hunting made me recognize that technology should be used on the trail to enhance a user's experience with nature rather than hindering it. In the future, I would research further on enhancing creativity and sharing capabilities within the application while ensuring that the application itself does not distract a user from his/her experience with nature in addition to potentially partnering with local state parks to field test the mobile application.

C1: GroupSafe

Arman Bahraini (arman1@vt.edu), Steffen Moseley (scm@vt.edu), Sungha Song (ssong716@vt.edu), Taiwen Jin (kim6@vt.edu), Zhizheng Chen (zc219@vt.edu)

Advisor: Dr. Tilevich (tilevich@cs.vt.edu)

GroupSafe is an application designed to provide connectivity between all the members in a group at all times in order to ensure everyone's safety. The goal of the application is to leverage current technologies available in modern smartphones to share and display locations on a map in real time. Users are able to create groups with unique names and passwords. Each group member shares his or her location and can chat with other members since websockets are utilized to provide a low overhead location and messaging system. In addition, the creator/host of the group can set up a radius which he/she expects everyone to be within at all time. If any member of the group is to wander past the radius set by the host, the application will send an alert to everyone in the group so they can react accordingly. For easy direct communication the app provides click to call or text. Google Maps is embedded in the app to show user locations, so in the event where a member does not respond to group chat, call, or text, everyone in the group will still be able to find the member by looking at the map. While the application was developed with college students and young adults as the primary clients in mind, other user bases may also find the app useful.

C2: A More Secured Pintos: Authentication and File Access Control

James Wong (phox2@vt.edu) Nathan Lane (lnathan8@vt.edu)

Advisor: Godmar Back (gback@cs.vt.edu)

Pintos currently has no means of handling authentication and managing file access control, which are necessary in order to have a secure operating system environment. A secure operating system requires distinguishing between multiple, possibly simultaneous, users and restricting readability, writability, and executability of files owned by different users. To do so would not only provide user level privacy, but it would also be beneficial for future, possible additions to Pintos, such as internet connectivity, where security becomes increasingly paramount. In terms of design, a passwords file an entry containing the username, encrypted password, group id and other pieces of associate data for each user. As of now, the SHA512 encryption method is used. A login program reads the authentication information typed in by the user. The password, when typed in, is encrypted using the SHA512 method. This encrypted input is compared with the encrypted password within the passwords file. Doing so would avoid needing to store passwords as unsecured plaintext anywhere on the Pintos machine. When the user logs in, the group id and user id are set in the shell process so that the operating system will be able to give the user the appropriate accesses to the appropriate files. In order to support this, each file and directory in the hard drive stores stores the user and group id of its creator along with the access permissions. Any attempt to access files would require checking such metadata against the effective user id and/or the effective group id set in the shell process. Currently, the login program is developed to handle input from users and to compare this input to information in the passwords file. The password file must be created the host machine housing Pintos and then imported into Pintos' file system. Each process has the potential to be identified to a particular user and group, along with the user and group identity through which it effectively executes commands. Each file that is created also takes in the creator's identifying metadata along with default permissions as defined by the kernel and a user-configured umask.

C3: Parking in Blacksburg

Jeongho Noh (jn0503@vt.edu), Tanya Sutan-Tanon (thanida@vt.edu), Daniel lm (dsim95@vt.edu), Tommy Hyres (tommy085@vt.edu), Teresa Lin (teresal@vt.edu)

Advisor: Dr. Andrea Kavanaugh (kavan@cs.vt.edu)

In the Blacksburg community, it can be difficult to find available parking due to unfamiliarity of the rules and unintuitive parking signs. People often get ticketed when they are unaware of the rules or misunderstand the parking sign. With the Parking in Blacksburg website, we aim to aid local Blacksburg residents, Virginia Tech students, and visitors to quickly find parking in downtown Blacksburg. This website will provide detail information, and exact location about the parking system in Blacksburg. In that way users can conveniently decide which parking lot they are going to use. During our design process we came up with questions such as: do we want a mobile website or an iOS/Android app?; what are the basic design elements of this application?; what does a basic user experience look like?; what mapping platform to use?, etc. We started our process by creating personas of our target user to get a better understanding of the application's requirements. We targeted visitors and those who are not familiar with the parking options and rules of Blacksburg. Additionally, we came up with different use cases that a user might have in order to come up with a list of problems for our application to solve. This allowed us to create a "minimum viable product" that can solve at least one of those problems and guided us in adding more functionality to resolve the additional problems. Our website shows a map of Blacksburg with parking lot information such as hours, parking pass requirement, price, etc., associated with each lot or area. Users are able to locate the closest parking options and use the information provided to select the most suitable. The Parking in Blacksburg website strives for ease of use as well as a responsive, interactive interface. Site functionality will include advanced searching and filter options. We will pass all progress made to Code for NRV for any further development. Currently this website does not provide parking information of the Virginia Tech campus, but should later be expanded to encompass all of Blacksburg. If this project is successful in aiding the community, future works could include a centralized parking system for Blacksburg that could later be expanded to surrounding areas.

C4: CS Shared

Meghan Hamannwright (meghankh@vt.edu(, Jodie Burnett (jodieb@vt.edu), Nitin Sudini (nsudini@vt.edu), Tanner Hudson (tannerh4@vt.edu), Usman Anwar (usmana94@vt.edu)

Advisor: Andrea Kavanaugh kavan@vt.edu

Our capstone project aims to solve an ongoing problem in the Virginia Tech organization, Computer Science Community Service (CS Squared). Speaking with site coordinators in CS Squared, concerns have been raised about what the most efficient and best way is to coordinate the visits in order to get every volunteer over to the site in a timely manner. Currently, the system involves simply sending an email out to the list serv letting members know of the upcoming visit and then planning a place and time to meet. While this system is simple, there can be complications. For instance, there have been instances where there are not enough cars to carry all the volunteers over; in some cases, there are no cars at all. There has also been no system in place to estimate about how many volunteers plan to show up. While speaking with site coordinators, it has been mentioned that sometimes they will get way more volunteers than needed for a visit. In one instance, ten volunteers showed up for a thirteen person elementary school class. Other problems that have arisen include volunteers not being able to contact their ride, or vice-versa, leaving volunteers with no way to reach the event. To solve this ongoing issue, our team decided to develop a web application to streamline the process of carpooling between volunteers. The application would allow users to sign up as a driver; they would be able to input information like number of seats, pickup location, and estimated time of arrival. Volunteers who need rides could use the application to see what seats are available, and reserve a spot if they need to. In an ideal scenario, this system would eliminate all communication problems between the driver and the passenger. In addition to the main functionality, there would be other materials available on the site as well, such as a schedule of upcoming events for the CS Squared club on the app's main page. We believe that this type of tool would make the process of volunteering for CS Squared much easier, and could encourage other potential volunteers to sign up. Many students might find the process of coordinating through email to be too slow, and this could drive volunteers away. By creating an easy-to-use and convenient web app that allows users to sign up for a ride at any time, engagement with community service could be increased.

C5: VR4GETAR

Jishnu Renugopal (jishnu@vt.edu), Mattin Zargarpur (mattinz@vt.edu), Christian Richardson (cjrich95@vt.edu), Will Schmidt (schmidtw@vt.edu), Haiyu Zhao (haiyuz93@vt.edu), Kevin Zhang (kyz@vt.edu)

Advisor: Dr. Edward Fox (fox@vt.edu)

Global Event and Trend Archive Research (GETAR) is supported by NSF (IIS-1619028 and 1619371) through 2019. It will devise interactive, integrated, digital library/archive systems coupled with linked and expert-curated webpage/tweet collections. This project will act as a supplement to GETAR by providing a Virtual Reality (VR) interface to visualize geospatial data and extrapolate meaningful information from it. It will primarily focus on visualizing tweets and images obtained from the GETAR data archive on a globe in a VR world. This will be accomplished using tools like Unity, HTC Vive, C# and Git. In order to ensure that the product meets the end user's specification, this project will use an iterative workflow with a very short feedback loop. The feedback obtained from Dr. Fox and our team members will be used to make subsequent prototypes and the final product. Our project is intended to be used as a demo by school children interested in data analytics and data sciences. Additionally, this project can also be extended to add features to our end product.

C6: Campus Marketplace

Ishita Ganotra (ishitag@vt.edu), Monica Wei (monicaw8@vt.edu), Tiffany Tran, (tiffany7@vt.edu), Will Schmidt (schmidtw@vt.edu)

Advisor: Andrea Kavanaugh, kavan@vt.edu

There is currently no means for college students to offer their skills and services in the campus community. Popular social media sites like Facebook do not provide a quick or reliable way for students to even search for specific services. Campus Marketplace is an integrated online platform where students can easily find or offer affordable services to practice their talents. Since it is exclusive to college students, the marketplace will have a positive impact on campus community, and help bring students together. It will allow students to log in with a valid ".edu" account, and each college will have their own platform. This not only creates a sense of safety and security, but also works to build a strong community through engagement, sharing, and trade.

C7: Building a Website for the Center for Research in SEAD Education

Paco Gallegos (pacog94@vt.edu), Tianyu Ge (tianyu@vt.edu), Victoria Hairston (vhairs16@vt.edu), Danna Kim (dannakim@vt.edu), Eunjee Kim (silverg@vt.edu)

Advisor: Andrea Kavanaugh, kavan@vt.edu

The goal for this project is to design and implement a new website for the Center for Research and SEAD Education. The Center focuses on inspiring PK-12 students, connecting the community to new opportunities, and impacting research in the fields of Science, Engineering, Arts, and Design education (SEAD). This website is made to target parents, educators, volunteers and researchers but does not limit content for other audiences. The development process involves frequent meetings with the clients and FourDesign, a faculty-led design company, to discuss styling and the content requirements of the website. While the design team is responsible for envisioning a website and critiquing our products, our job is ultimately to develop a functioning website to present to the client. After discussing textual information, images, color schemes, and layout, we began implementing a WordPress website, which resides on a development server. Using WordPress as a content management system, we utilized its themes, plugins and simple interface to suit the Center's typical use cases. Extensive user testing was done on the website to enhance the overall experience and improved accessibility to accommodate varying demographics. Backed with user testing results, we also went one step further and proactively provided the Center with consulting, which helped streamlining the user interaction flow. The long term goal of this project is to establish an engaging platform which can demonstrate the Center's value propositions to their target audience.

C8: Networking in Pintos

Alexander G. Glasson (aglasson@vt.edu), Marcus Wanner, Marcus (marcusw@vt.edu), TJ Corley (tjcorley@vt.edu)

Advisor: Godmar Back

Pintos recently has started the transition to becoming a more modern operating system. For example SMP was just introduced this year. One feature that any modern operating system has is networking and every device now is able to communicate with the outside world. However pintos sorely lacked any networking capabilities. Our group(Alex Glasson, TJ Corley, and Marcus Wanner) had an interest in this space and thought adding support for networking in Pintos would be a challenging task that could truly be useful. Broadly, all that is needed for networking is: A driver for the network card that will physically send and receive packets, logic to create and process packets, and a way for userspace to interact with this kernel networking component. We decided to implement the driver for the Intel E1000 card. This is because it is the default network card for QEMU, and there is extensive documentation online describing the behavior of the card. For the packet processing we leveraged LwIP. LwIP, which stands for Lightweight IP, is a very low resource TCP/IP protocol suite that is adaptable to almost any system. Finally, for the userspace component, we decided to mimic the UNIX socket syscall API. This allows us to port UNIX networking applications, such as a web server, with minimal work. Currently, we have a fully working E1000 driver that properly initializes the card and is able to send and receive packets. An interesting implementation aspect of the driver is the fact that PCI interrupts are used to determine when to pull packets from the card rather than a polling thread. We also have created the relevant syscalls that userspace programs will need, and are currently writing programs that test their functionality. These include basic programs that simply open and close sockets, all the way to a HTTP web server. The LwIP integration is still in progress. The two major aspects are integrating LwIP with our build system, and providing data structures and function implementations that the core of LwIP depends on. This aspect of the project has been more challenging than we anticipated.

C9: Piping in Pintos

Andrew Williams (dwill225@vt.edu), Aaron Rivenberg (arivenb@vt.edu)

Advisor: Godmar Back (gback@cs.vt.edu)

Our Project involved the addition of pipes and I/O redirection to the Pintos operating system. In Pintos child processes do not inherit the file descriptors of their parent processes. Consequently, processes in pintos are not able to easily communicate with one another with ease. Adding these features allows for parent processes to send input or receive output from child processes. It also enables child processes to be able to send their output to other child processes or files. Our implementation focuses on the use cases of shells. When a pipe is created, two file descriptors are returned to the user: one for reading, one for writing. These file descriptors can then be read from and written to in the same way that any normal file would be, as long as only reading occurs from the read end and writing from the write end. Processes also have the ability to pass up to 2 file descriptors of any kind to their children. These files or pipes will be removed from the parent's table and set as the standard input or output of the children. This enables processes to read from and write to files, but have it appear as though they are reading from and writing to standard input and output as usual. With this implementation we are able to run a simple shell that supports both piping and redirection. Currently our tests focus reading and writing from pipes with multiple size inputs and different read and write combinations. Also, we have created a few programs that we can run in our shell to demonstrate the communication between processes. We plan to write more tests that will show the decrease in execution time when the reading and writing process run in parallel.

C10: Mr. Steal Yo Election

Divyansh Gupta (divyg@vt.edu), Zakk Lefkowits (zakkl13@vt.edu), Mikias Alemu (mikias@vt.edu), Bhaanu Kaul (bhaanuk5@vt.edu)

Advisor: Siddharth Krishnan (siddkris@vt.edu)

This project discusses the correlation between sentiment and political affiliation based on Twitter tweets around the United States. Since the creation of billion user social media networks, it has become increasingly easy for users to share their opinions and feelings to the world. This has triggered in the over flooding data throughout the internet. Now called Big Data, the vast amount of information available on the internet is becoming increasingly difficult to analyze due to its sheer amount and thus being costly in resources. This project focuses on harnessing the power of a vast microblogging network, Twitter, to gain insight into the political leanings of the American people. Traditionally polling has been used to achieve the same goal, yet polling has many shortcomings - notably the immense cost to reach a large audience. Twitter bypasses this by providing simple access to billions of users' thoughts, opinions, feelings, etc. By applying modern natural language processing and machine learning techniques to Twitter data, we are able to produce quantitative data on the U.S. political feelings of users in America and around the world. The core of our work begins in sentiment analysis and a custom built model to classify text as leaning toward the Republican, Democratic, or third parties. We used this system to process over 3 million tweets from the 2012 campaign season. Through statistical analysis on our processed data, we developed a model to predict where each state's electoral college votes will fall, based purely on Twitter data. The model is designed to be applied to future elections.

C11: Nui

Bridget Larcher (blarcher@vt.edu), Jordan White (jordanrw@vt.edu), Rupin Khera (krupin5@vt.edu)

Advisor: Professor Steve Harrison

Nui aims to create new user interfaces through extreme simplification. Since the first computer, developers and designers have been changing and optimizing user interfaces through the shift from purely text to the addition of colors, graphics, icons and transitions. Our goal is to re-imagine the norm of user interfaces and push beyond the dominant use of text and icons. Nui explores a new future of UI through colors, shapes and transitions.

Our presentation includes different user interface prototypes, including weather and navigation apps. We have two interactive prototypes as well as sketches and mockups of other ideas.

C12: Multithreading Support for PintOS

Alex May (amay985@vt.edu) Sean Rigden (sarigden@vt.edu) Ha Nuel Lee (hanuelee@vt.edu)

Advisor: Godmar Back (gback@vt.edu)

Parallelism is becoming an essential part in modern user applications. User applications such as multiple client-supporting web servers and event-driven mobile applications utilize concurrency to handle multiple tasks asynchronously. With the increasing popularity of parallelism, it is expected that most present-day operating systems support multi-threading. PintOS is a small-scale operating system for x86 processors that we built in this course. PintOS supports multiple processes running on multiple CPUs, a lightweight set of system calls and virtual address space, but lacked the ability to support multiple threads within a process. This project expands on this base implementation of PintOS by consolidating resources shared between all threads to a process control block. This includes a process's virtual memory space including the program's executable file, open file descriptors, and any mapped memory regions. A framework was made where users can use system calls to spawn new threads within a process. To utilize multithreading, user level synchronization is necessary. This project implements a very basic form of Linux's futex API to build lightweight mutexes, semaphores, and condition variables in user space. The user level API for these synchronization primitives follows the POSIX Threads library API, but is much less robust in its implementation. One of the main challenges in user level synchronization is the cost of switching into kernel mode. By utilizing futexes and enforcing an ordered sequence of actions between threads, we were able to allow certain synchronization situations to remain entirely on the user side. This optimizes the cost of many common case situations such as acquiring an uncontended lock. The current implementation supports running multiple threads, protecting critical sections, and allocating dynamic memory. These features were tested by integrating projects from the Computer Systems course, one of which is a threadpool. A threadpool is a framework to support task-parallel applications using a fork/join model. The operating system works consistently when running these projects. Utilizing multiple threads shows an improvement in run-time for various tests within PintOS. Running the quicksort algorithm on an unsorted 80 MB buffer on eight CPUs in parallel showed roughly a 100% speedup compared to a single threaded run.

C13: How News Appears on Social Media: An Analysis Comparing Reputable News Sources to Twitter and Reddit Trends

Morgan Mazer (morganm7@vt.edu), Krista McGuigan (kmcguig3@vt.edu), Brandon Ray (branr94@vt.edu), Michael Mahedy (mahedy@vt.edu)

Advisor: Siddharth Krishnan (siddkirs@vt.edu)

In today's society, the power of social media is rising. It is becoming increasingly common for people to be active on a variety of social media sites and depend on these sites for not only socialization, but also for news. As many people are informed about news topics worldwide from social media exclusively, we are curious to see what types of news are appearing on social media and how they are being talked about. We are also interested to see if the news that appears on certain social media sites has a set of common topics and themes to give insight into the user demographic and what type of news commentary could be expected from different sources. This project involved scraping data from Twitter, Reddit, reliable news sources, and Google trending topics to determine what people are talking about on major social media sites and compare it to what's going on in the world. Our analysis aims to find out what people care about, what they don't (or may not know about), and in what ways they care. To do this, we used data collected from news sources and Google trending topics as a baseline to compare to posts on popular Reddit pages and trending hashtags on Twitter.

C14: Validating Growth Models for Concurrent Cascades in Social Networks

Rohan Kaul (rohan@vt.edu), Henry Steffensen (henry95@vt.edu), John Purviance (ffdd4846@vt.edu)

Advisor: Siddharth Krishnan (siddkris@cs.vt.edu)

Widespread adoption of social networking sites like Facebook and Twitter is changing the way that people communicate and spread ideas. Today, many people receive information about events and global news over these social networks. Usage of these networks has expanded to event planning and advertising as well. For law enforcement, it is important to understand the turnout for gatherings managed on Facebook and Twitter. Similarly, advertising agencies are interested in understanding the impact that marketing campaigns have by monitoring consumer discussion. The work put forth in "Analyzing the Growth of Concurrent Cascades: A Forest of Trees Approach" by Krishnan and Ramakrishnan is a framework simulating hashtag propagation over Twitter. To expand upon that work, we have implemented the principles from that work to generate and analyze synthetic Twitter networks. Our work simulates the spread of hashtags that are propagating over Twitter from internal and external sources. These simulated networks grow by using the Chinese Restaurant Process (CRP) and Preferential Attachment (PA) to join links between users retweets and replies. As the network grows a forest of trees that model hashtag propagation emerges. This forest, an Information Cascade, is analyzed against data from Twitter to validate network properties. Our investigation focuses on the 2017 Oscars, London Parliament Attack and the Higgs Boson Particle.

C15: Pintos Multithreading

Colin Kirkpatrick (kcolin7@vt.edu), Joe Palazzo (pjoe3@vt.edu), Ryan Mayer (ryanmm99@vt.edu)

Advisor: Godmar Back

Multithreading is an integral aspect of modern computing for both big data computing and increasingly complicated user end programs. For this reason we implemented a multithreading system in the Pintos OS. We implemented POSIX style pthreads modeled after the Linux implementation. This gives the user access to true multithreading which is managed by the OS as opposed to a user level design. A kernel level implementation allows for the OS to take full advantage of multiple CPUs, and allows the user to execute a single program with multiple concurrent tasks. In order to determine the benefits of parallelization, one can use Ahmdal's law, which determines that the overall 'speedup' of a program running in parallel, verse a serial program, is determined by the fraction of program that is parallelizable (P), versus the fraction that is serial (S), as well as the number of available CPUs (N). The equation is given as : 1/((1-P)+(P/N) As our implementation is done at the kernel level, the kernel is in charge of maintaining the abstraction of processes and threads. The base Pintos kernel allows for threads to exist in a one-to-one fashion with user programs, for each user program that runs, the kernel manages one associated thread. The multithreaded Pintos kernel allows for threads to be separate from processes, and thus adds a layer of management between process and thread level information. This separation of data allows for multiple user generated threads to work on the same set of data within a process, allowing the user to better utilize their own systems CPU when running computation heavy programs which can be executed in parallel. Thread scheduling and memory management is handled natively by the OS, which allows for users to focus on their own programs implementation, without worrying about the specifics of how an OS's multithreading implementation. A user program will, however, have to rely on use of synchronization devices, which will be exposed via the user side multithreading API. Ultimately, multithreading should allow for users to have full access to creating and running their own threads, and giving them the capabilities to do so via synchronization tools.

C16: Implementing User Controls and the Bell-LaPadula Access Model in PintOS

Zachary Burch (zchryb@vt.edu), Matthew Stevens (smatth25@vt.edu)

Advisor: Godmar Back (gback@vt.edu)

Operating system security historically has had multiple issues in both implementation and design. PintOS has no form of security, which has allowed our group to have a clean slate to design and implement the Bell-LaPadula access control model, a state transition security model that ensures data confidentiality. The decision on this model was to help understand the requirements and restrictions for alternative system security policies, which will be useful when working with modern systems in the future. In the process of achieving the goal of an access control model, we implemented the user abstraction, where multiple processes are all bound together by one user identifier. Then we used the Bell-LaPadula model to restrict how users could manipulate files in the system. To further provide security for the system, we implemented basic authentication by requiring a username and password for users. Finally, the passwords are hashed using SHA256 and stored on the file-system with the highest level of security possible. The differences between the standard Unix discretionary model and the Bell-LaPadula model have broadened our understanding of information security models.

C17: Implementing TCP/IP networking for pintOS

Kara Vaillancourt (kvaillancourt@vt.edu), Kate Nguyen (katevy@vt.edu), Adam Barth (adam797@vt.edu)

Advisor: Godmar Back (gback@vt.edu)

Network connection is associated with most of the modern activities for which computers are used. Even ignoring the obvious cases of browsing the web, applications check online for updates, files get backed up to the cloud, and a large amount of other functionality depends on being able to communicate on a network. Thus, the need for networking in any kernel is immediately apparent, yet the Pintos operating system, in its current form, has no support for it. For our project, we added networking functionality to Pintos by writing a driver for the E1000 network interface controller and porting the open source lwIP TCP/IP stack into the operating system. User programs can make use of the addition of networking by BSD-style system calls that interact with lwIP. We demonstrate the networking capability of Pintos by running a simple web server and accessing it from a local machine, exhibiting its ability to interact with networks at large.

C18: Efficient Multithreading in Pintos on KVM

Eric Williamson (ericrw96@vt.edu), Chris Wu (cjwu95@vt.edu), Peter Steele (peter707@vt.edu)

Advisor: Godmar Back (gback@vt.edu)

The Pintos kernel originally only supported running one thread per user process. Implementing efficient multithreading in Pintos is a useful exercise to understand the tradeoffs inherent in operating systems that allow processes to run in multiple threads. First off, we had to distinguish between what each thread in a process should own in their own thread control blocks and what they should share in their process control block. Joining threads, sleeping threads, and exiting threads all required careful consideration of key data structures in the kernel. Exiting blocked and sleeping threads is especially tricky, because kernel data structures must be not be left in inconsistent states. Several previous Pintos design choices which assumed a single thread per process had to be modified. For user programs, we provided efficient user-level synchronization primitives via futexes, mutexes, and semaphores as well as user-level dynamic memory allocation. We were able to run several multithreaded user programs in Pintos running on KVM with correct results and show considerable speedup with an increasing number of threads.

C19: College Football Rankings Multiple Linear Regression Model

Grant Hughes (granth6@vt.edu), Kyle Johnson (kylej13@vt.edu), Lucas Rose (lucasr95@vt.edu)

Advisor: Siddharth Krishnan (siddkris@vt.edu)

Deciding college football rankings has always been a source of fierce debate. A committee provides the official rankings using a variety of factors, however, the process contains a high level of subjectivity. The goal of this study is to objectify the selection process by developing a model to produce rankings. Machine learning is used to build the model, which is fit with past game data and rankings. This study highlights that in the world of big data and high-level processing, even complex problems such as deciding college football rankings can be solved objectively.

C20: Fake News and Social Media

Margaret Stump (marstump@vt.edu), Wilson Rhodes (rush@vt.edu), Luis Villavicencio (lv23@vt.edu), Alex Lu (alex07lu@vt.edu)

Advisor: Siddarth Krishnan (siddkris@cs.vt.edu)

Through our research, we are trying to understand the propagation of fictitious news stories on social media. Over the past few years, false news stories have spread widely and rapidly on Facebook and other social media websites under the guise of being true. This is due to the lack of fact checking by users and the social media sites themselves. The spread of false information leads to a misinformed population and reduces the quality of political and social discourse within society. We are trying to find out why and how fake news can spread so quickly, and what could be an effective way to stop the spread of fake news and falsehoods on social media sites.

C21: Social Sentiment Using Stock as Lens

Jake Hendrzak (hendo21@vt.edu) Jisu You (silven@vt.edu) Haitham Mawari (haitham1@vt.edu) Pejman Anbai (peji@vt.edu)

Advisor: Siddharth Krishnan, siddkris@cs.vt.edu

Twitter hashtags allow users to highlight their tweet to focus on a certain cause or topic. In the recent years, hashtags have included companies along with a sentiment to express the user's feelings toward said company. Our group has implemented an algorithm that reads through tweets and predicts the change in the company's stock price based on the sentiment of the tweets. By parsing each word of each tweet, our algorithm can gather a sentiment value for the tweet based on each word used. With the calculated sentiment value stored, the algorithm then predicts the change in stock price that the company will experience. This project allows us to analyze the impact that social media and hashtags has on companies and movements. Twitter users will be able to analyze the impact that they can have through social media movements, as well as learn from past movements to benefit their own.

C22: Escape Room

Patrick Folena (pfolena@vt.edu), Burhan Ishaq (iburhan@vt.edu), Se Eun Park (katee93@vt.edu), So Hyun Jo (sohyun@vt.edu)

Advisor: Dr. Scott McCrickard

This project explored ways to develop a Virtual Reality game designed to give users a social gaming experience. The experience focuses on fostering cooperation within small groups to exercise their minds within a 3D environment. Escape room is a place where players enter a room with puzzles and the players have a time limit to solve a series of
puzzles that will help them find a key that unlocks the door.

Our project wanted to focus on the team element, so we designed Escape Room to require two people to play interactively and help each other. The puzzles will be split across the room, and two users have to communicate to get the answer. The game room has 4 questions displayed on the wall, and two players have to solve them to escape the room. The players have a limited time to solve the puzzles and can compete with other teams for the best time. We utilized an Android phone to install the game package and directly test the game. Using a Google cardboard headset, we could experience the game personally and ensure that the users receive our intended experience. We tested the functionality of button presses with target users on the keypad for connection, correspondence, and transmission with the light board that indicates the state of correctness.