SociOS

An Operating System for wearable AI devices

Authors
Riley Miyamoto,
&
Christian Nalumen

ICS 240

Introduction

For as long as human beings have roamed the earth, the desire for a companion was always there, where it be a fellow person, a loyal pet, or even an imaginary friend. Throughout the ages, mankind have made countless efforts to improve the quality of life. The greeks used hydraulic and pneumatic systems to build autonomous machines for entertainment and curiosity. During the Industrial Revolution, machines, steam engines, and many more innovations were made because there was a need to increase production. Now, we have computers, cell phones, and tablets because there is a desire to stay connected with each other. Because of these humanly desires, it’s only natural that we look at Artificial Intelligence.

Preface

The Idea of AI, or Artificial Intelligence is not anything new. It’s been around for centuries in the form of myths and legends. In Greek mythology, it is said that Hephaestus built autonomous servants. The movies have depicted AI in numerous forms: humanoid cyborg/android(T-1000, iRobot, DATA), sidekick car Interface(K.I.T.T.), smart house interface(PAT), runaway secret military weapon (Johnny 5), and assistant to Tony Stark (JARVIS). In the real world, we have supercomputers that can play Chess and Jeopardy, and most recently, SIRI, which acts a personal assistant through speech interaction. Now, we have Smart Watches, Google Glass, and Nike+, it’s safe to say that there is a growing trend in wearable devices. With cpu and memory improving in speed and power while decreasing in size, a wearable device with an AI interface may soon be possible, and eventually have a system like JARVIS that regular consumers can afford. A Smart Wearable AI device will not only become an extension to mobile devices, but also an extension to ourselves.

1.0 Hardware

1.1 Overview

Before getting to the OS, let's establish the hardware used as a base to create this Operating System. The focus was to create a system that is very energy efficient, yet powerful enough to run an AI system.

1.2 Processor

Firstly, the processor chosen for this is the Cortex-A53 which is based of ARMv8-A technology. This processor will most likely be in a lot of mobile devices by 2014. It is a quad-core processor and is capable of supporting 32 bit and 64 bit instruction sets. This was primarily chosen because of it's speed and energy efficiency. To control memory, the CoreLink CCI-400 was chosen because it was made to work side by side with the Cortex-A53.

1.3 Memory

For main memory, an LPDD3 or LPDDR4 would be prefered because again, it's power efficient. The LPDDR3 may also be up to 3gb of memory. A built-in 8-16gb or more flash memory would be used for the OS. A microSD slot would allow the device to have expandable memory.

1.4 Input/Output

As for I/O, there will be a built-in mic and speakers for speech recognition and speech synthesis. An OLED touch screen will add vibrant colors and flexibility. For connectivity, it will have bluetooth to connected to mobile devices to view text messages and caller notification, wifi to receive email, download apps, connect to social media, etc. It may also be equipped with other sensors such as GPS, heart-rate monitor, electromagnetic compass to track a user's fitness progress.

2.0 Operating System

2.1 Overview

In order to have Artificial Intelligent devices, there must be intelligent operating systems. The operating system will need to adapt to the user's needs and personality in order to become the ideal companion. *SociOS*, which name derives from the latin word, socio, meaning companion, will be the driving force for this Smart Wearable AI device. SociOS will be off the Linux kernel. The reason being is because Linux offers three main features desired for a project like this: affordability, customizability, and scalability. For this particular device, the main target market would be geared toward the average consumer, so the end price would have to look attractive enough buy. Using a Linux-based kernel would drive the cost down because of the open source nature of it, there would be less licensing and royalty fees. In order for the AI Interface to adapt to the user, the system will have to be very customizable in order to meet the needs of the user. And lastly, when creating a wearable device, it would have to be light and compact, and so must the operating system. For example, a heavy GUI desktop environment such as Gnome or KDE would not be necessary, but rather a light weight since the internal flash memory would only be anywhere between 8gb to 16gb.

2.2 Users: AI Mode

In this device, there will be two groups: AI and Human. The AI Interface will be treated as another user so the operating system would be a multi-user system. The reason behind that is because the AI interface would have the ability to read, write, or execute files just like any other user would. Its decision making would be based on a complex series of algorithms. The AI User would make decisions based on previous data collected, real-time data, and make predictions to choose the appropriate or best course of action. It would read data from various input devices or sensors such as the microphone, GPS, accelerometer, etc. Upon initial startup, The AI interface would ask the Human User to set up a cloud storage service to sync with, and then ask a series of questions such as favorite color and store the data to use for future reference. As the time goes by, the AI interface would update the stored profile data to make better decisions when interacting with the Human User. While the device is charging and connected to the internet, the AI Interface would dump a copy of the collected data into the cloud storage as it updates and overwrites old data. The memory stored in the cloud would be known as "Cognitive Memory".

2.3 Users: Human Mode

In Human User mode, the user would access the OS through a GUI. This use would be able to access the operating system just like any other mobile device. The device would have the common input device/buttons such as a touch screen, power button, volume rocker, home button, back button, and even an AI Button. When the AI Button is pushed, it invokes the AI interface, and the interface will listen for speech commands. A long press would activate real-time mode, where the AI stays on to record real-time data; this feature can be used to track fitness progress, vital signs, and the AI will be always listening without pressing the button again. The Human User will also be able to access the command line through proprietary or third-party software when directly connected to a computer via usb or bluetooth. Users will . In the command line, the user will be able to login as Human or AI. This feature is to allow users to configure files, application, process management settings, etc. However, debugging mode will have to be enabled.

3.0 Kernel

3.1 Kernel Overview

Because this will be made for smaller devices, a lighter kernel will be needed. For this reason, a *Microkernel* type was used. Since the kernel acts as an intermediate devices drivers and application, it will is responsible for how the A.I handles I/O devices and functions such as Speech recognition, speech synthesis, and information gathering. While in AI mode, speech data will be parsed and analyzed in order for the AI to understand what the Human User is requesting. It will also manage text-to-speech so email, text, and notifications may be read to the Human User.

4.0 Process Management.

4.1 Overview

In this type of AI system where it acts as an intermediate between the kernel and the Human User. in most cases the user will always receive the highest priority upon request. For example, when the User presses the AI button, the current process will be switch to the waiting state and the AI interface will be brought into the running state, then once terminated, the previous process will be brought back to the running state.

4.2 AI Control

During AI mode, what ever request the Human User gives to the AI will spawn as child processes under the AI. With this the interface is able to control a process. If it finds a process unstable or, it may put that process to a block or suspend state. It will act as an Interactive User to improve efficiency. To improve performance, the AI may also adjust the processor affinity based on the tasks of the processes waiting to run.

4.3 Interrupts

The interrupt handler will handle interrupts in different manner depending on the priority levels. For hardware, only the AI button will not be ignored. The reason for this is because the purpose for the AI is to meet the needs of the user. The only exception is when the OS is in the halt or boot sequence, all hardware interaction will be ignored . Software interrupts will also be handled on a priority base such as if the device battery hits a critically low, it will switch the current process to the wait/suspend state, and put the AI interface in the running to inform the user that the battery level is low and must take immediate action, then it will save the changes that had been made in case of the battery dying. Also, processes will have the highest priority if it has to make a deadline.

4.4 Scheduling

To address scheduling, the AI will use an Interactive scheduling system. Multilevel feedback queue will be as the Scheduling algorithm because it ties in with the functions of the AI interface since it would already be an Interactive User. It would improve turnaround time, response time, throughput. It a process has a higher priority, it would enforce it. Threads will be dynamically scheduled by the AI in order to improve performance.

4.5 Deadlocks

In the case of deadlocks, prevention and avoidance will be key. In order to prevent deadlocks, the bankers algorithm will be implemented into the AI. This would be good because it would take advantage of the AI’s ability to analyze processes/threads. In order to recover from a deadlock, as a last resort if the AI would not be able to release a process from a resource, it would release itself from the cpu.

5.0 Memory Management

5.1 Overview

In this type of Operating System, it is key to have memory being written and accessed at appropriate speeds. A lot of the different types of memory management will be based around dynamic because we wanted memory to be written at the moment it needs to be, to be more efficient. It is also heavily priority based.

5.2 Structure Requirements

There were five requirements when creating the memory management. These five requirements were relocation, protection, sharing, logical organization and physical organization. For relocation memory is allowed to be shared among processes, when a higher priority level process needs it. Addressing is based upon the level of priority and the functionality of the processes. When a request comes from a higher priority, it will be able to write to memory and after it has finished the prior process can continue. Sharing memory will be allowed only when higher priorities need to finish. Logical and Physical Organization will be segmented to organize the different levels of priority. This will make it easier to locate the different priorities when needed.

5.3 Partitioning

The Operating System will be using dynamic partitioning to help utilize all blocks of memory. Although there may be external fragmentation while using a dynamic partition, it is more important that we optimize as much memory as possible to make sure that the operating system is utilizing the memory it has.

5.4 Paging Algorithm

The paging algorithm that is going to be used is the Least Recently Used (LRU). The reasoning behind this is because the operating system is an AI there are function that may not be used as much as others depending on others. By using LRU we can take out pages that don't get used and replace them with ones that are. This allows for the OS to keep changing based on what the user is doing.

5.5 MMU

To help memory management there are chipsets that work to optimize it. These chipsets are the CoreLink MMU 400, this was built to work with the processor to we are using. It has fault handling, logging and signaling. It also has debugging and performance monitoring. The CoreLink GIC-400, NIC-400, DMC-400 and CCI-400 also work to help to make it more efficient and better at power saving.

5.0 File Management

5.1 Overview

The file system management, like the operating system will have to be able to learn based on the user. The types of files that are used within the OS will depend on what kind of user is using it. Files that are more commonly used will alter the design to better suit the user. All file types are used since the OS was built for all users.

5.2 File Organization and Access

The file organization and access uses the Direct/Hashed file. This organization and access type allows for direct access to known address space. The OS will sometimes have to grab files at a moments notice so we needed a way for it to be able to do so. The Direct/Hashed file works great with this and it will help with it's performance.

5.3 File Directories

The file directory that will be used is a hierarchical directory. This directory allows for files to be group and allows for each user to have their own private root directory. This helps to stop others from going into your device and viewing files and to keep different users devices seperate.

5.4 Access Right and Record Blocking

User are allowed knowledge rights, but nothing else. To help keep the OS personal to each user, we can't allow others to alter or view the files. We will be using a fixed record blocking because it helps with I/O, which is what this OS will be working most with.

5.5 File Allocation and Free Space Management

The allocation that will be used is dynamic allocation since it is more reliable than pre-allocation. it will also be used with Indexed Allocation, so retrieving the file will be easier since it is indexed. We will be using a bit table since it does work well with any file allocation method, but also because it is as small as possible so it won't use extra free space that we wouldn't want it using.

References:

"Learn How to Use Electronic Sensors." *Ladyadanet Blog RSS*. Np, 27 Apr. 2012. Web. 16 Dec. 2013.

Wei, Wang. "J.A.R.V.I.S : Artificial Intelligence Assistant Operating System for Hackers." *The Hacker News*. The Hacker News, 5 Sept. 2013. Web. 16 Dec. 2013.

Grisenthwaite, Richard "ARMv8 Technology Preview."
http://arm.com/images/Cortex_A53_large.png

Damato, Joe. "Threading Models: So Many Different Ways to Get Stuff Done. at Time to Bleed by Joe Damato." *Time to Bleed by Joe Damato RSS*. Np, 8 Oct. 2008. Web. 16 Dec. 2013.

Apple. "Apple - IOS 7 - Siri." *Apple - IOS 7 - Siri*. Apple, 2013. Web. 16 Dec. 2013.

"Tutorials Point - Simply Easy Learning." *Operating System Memory Management*. Np, 2012. Web. 16 Dec. 2013.

ARM. "CoreLink System Memory Management Unit." - *ARM*. N.p., 2013. Web. 16 Dec. 2013.

<Legal Notices to be attached when OS is made>

GNU GENERAL PUBLIC LICENSE
Version 2, June 1991

Preamble

The licenses for most software are designed to take away your freedom to share and change it. By contrast, the GNU General Public License is intended to guarantee your freedom to share and change free software--to make sure the software is free for all its users. This General Public License applies to most of the Free Software Foundation's software and to any other program whose authors commit to using it. (Some other Free Software Foundation software is covered by the GNU Library General Public License instead.) You can apply it to your programs, too.

When we speak of free software, we are referring to freedom, not price. Our General Public Licenses are designed to make sure that you have the freedom to distribute copies of free software (and charge for this service if you wish), that you receive source code or can get it if you want it, that you can change the software or use pieces of it in new free programs; and that you know you can do these things.

To protect your rights, we need to make restrictions that forbid anyone to deny you these rights or to ask you to surrender the rights. These restrictions translate to certain responsibilities for you if you distribute copies of the software, or if you modify it.

For example, if you distribute copies of such a program, whether gratis or for a fee, you must give the recipients all the rights that you have. You must make sure that they, too, receive or can get the source code. And you must show them these terms so they know their rights.

We protect your rights with two steps: (1) copyright the software, and (2) offer you this license which gives you legal permission to copy, distribute and/or modify the software.

Also, for each author's protection and ours, we want to make certain that everyone understands that there is no warranty for this free software. If the software is modified by someone else and passed on, we want its recipients to know that what they have is not the original, so that any problems introduced by others will not reflect on the original authors' reputations.

Finally, any free program is threatened constantly by software patents. We wish to avoid the danger that redistributors of a free program will individually obtain patent licenses, in effect making the program proprietary. To prevent this, we have made it clear that any patent must be licensed for everyone's free use or not licensed at all.

The precise terms and conditions for copying, distribution and modification follow.

GNU GENERAL PUBLIC LICENSE

TERMS AND CONDITIONS FOR COPYING, DISTRIBUTION AND MODIFICATION

0. This License applies to any program or other work which contains a notice placed by the copyright holder saying it may be distributed under the terms of this General Public License. The "Program", below, refers to any such program or work, and a "work based on the Program" means either the Program or any derivative work under copyright law: that is to say, a work containing the Program or a portion of it, either verbatim or with modifications and/or translated into another language. (Hereinafter, translation is included without limitation in the term "modification".) Each licensee is addressed as "you".

Activities other than copying, distribution and modification are not covered by this License; they are outside its scope. The act of running the Program is not restricted, and the output from the Program is covered only if its contents constitute a work based on the Program (independent of having been made by running the Program). Whether that is true depends on what the Program does.

1. You may copy and distribute verbatim copies of the Program's source code as you receive it, in any medium, provided that you conspicuously and appropriately publish on each copy an appropriate copyright notice and disclaimer of warranty; keep intact all the notices that refer to this License and to the absence of any warranty; and give any other recipients of the Program a copy of this License along with the Program.

You may charge a fee for the physical act of transferring a copy, and you may at your option offer warranty protection in exchange for a fee.

2. You may modify your copy or copies of the Program or any portion of it, thus forming a work based on the Program, and copy and distribute such modifications or work under the terms of Section 1 above, provided that you also meet all of these conditions:

a) You must cause the modified files to carry prominent notices stating that you changed the files and the date of any change.

b) You must cause any work that you distribute or publish, that in whole or in part contains or is derived from the Program or any part thereof, to be licensed as a whole at no charge to all third parties under the terms of this License.

c) If the modified program normally reads commands interactively when run, you must cause it, when started running for such interactive use in the most ordinary way, to print or display an announcement including an appropriate copyright notice and a notice that there is no warranty (or else, saying that you provide a warranty) and that users may redistribute the program under these conditions, and telling the user how to view a copy of this License. (Exception: if the Program itself is interactive but does not normally print such an announcement, your work based on the Program is not required to print an announcement.)

These requirements apply to the modified work as a whole. If identifiable sections of that work are not derived from the Program, and can be reasonably considered independent and separate works in themselves, then this License, and its terms, do not apply to those sections when you distribute them as separate works. But when you distribute the same sections as part of a whole which is a work based on the Program, the distribution of the whole must be on the terms of this License, whose permissions for other licensees extend to the entire whole, and thus to each and every part regardless of who wrote it.

Thus, it is not the intent of this section to claim rights or contest your rights to work written entirely by you; rather, the intent is to exercise the right to control the distribution of derivative or collective works based on the Program.

In addition, mere aggregation of another work not based on the Program with the Program (or with a work based on the Program) on a volume of a storage or distribution medium does not bring the other work under the scope of this License.

3. You may copy and distribute the Program (or a work based on it, under Section 2) in object code or executable form under the terms of Sections 1 and 2 above provided that you also do one of the following:

a) Accompany it with the complete corresponding machine-readable source code, which must be distributed under the terms of Sections

1 and 2 above on a medium customarily used for software interchange; or,

b) Accompany it with a written offer, valid for at least three years, to give any third party, for a charge no more than your cost of physically performing source distribution, a complete machine-readable copy of the corresponding source code, to be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,

c) Accompany it with the information you received as to the offer to distribute corresponding source code. (This alternative is allowed only for noncommercial distribution and only if you received the program in object code or executable form with such an offer, in accord with Subsection b above.)

The source code for a work means the preferred form of the work for making modifications to it. For an executable work, complete source code means all the source code for all modules it contains, plus any associated interface definition files, plus the scripts used to control compilation and installation of the executable. However, as a special exception, the source code distributed need not include anything that is normally distributed (in either source or binary form) with the major components (compiler, kernel, and so on) of the operating system on which the executable runs, unless that component itself accompanies the executable.

If distribution of executable or object code is made by offering access to copy from a designated place, then offering equivalent access to copy the source code from the same place counts as distribution of the source code, even though third parties are not compelled to copy the source along with the object code.

4. You may not copy, modify, sublicense, or distribute the Program except as expressly provided under this License. Any attempt otherwise to copy, modify, sublicense or distribute the Program is void, and will automatically terminate your rights under this License. However, parties who have received copies, or rights, from you under this License will not have their licenses terminated so long as such parties remain in full compliance.

5. You are not required to accept this License, since you have not signed it. However, nothing else grants you permission to modify or distribute the Program or its derivative works. These actions are prohibited by law if you do not accept this License. Therefore, by modifying or distributing the Program (or any work based on the Program), you indicate your acceptance of this License to do so, and all its terms and conditions for copying, distributing or modifying the Program or works based on it.

6. Each time you redistribute the Program (or any work based on the Program), the recipient automatically receives a license from the original licensor to copy, distribute or modify the Program subject to these terms and conditions. You may not impose any further restrictions on the recipients' exercise of the rights granted herein. You are not responsible for enforcing compliance by third parties to this License.

7. If, as a consequence of a court judgment or allegation of patent infringement or for any other reason (not limited to patent issues), conditions are imposed on you (whether by court order, agreement or otherwise) that contradict the conditions of this License, they do not excuse you from the conditions of this License. If you cannot distribute so as to satisfy simultaneously your obligations under this License and any other pertinent obligations, then as a consequence you may not distribute the Program at all. For example, if a patent license would not permit royalty-free redistribution of the Program by all those who receive copies directly or indirectly through you, then the only way you could satisfy both it and this License would be to refrain entirely from distribution of the Program.

If any portion of this section is held invalid or unenforceable under any particular circumstance, the balance of the section is intended to apply and the section as a whole is intended to apply in other circumstances.

It is not the purpose of this section to induce you to infringe any patents or other property right claims or to contest validity of any such claims; this section has the sole purpose of protecting the integrity of the free software distribution system, which is implemented by public license practices. Many people have made generous contributions to the wide range of software distributed through that system in reliance on consistent application of that system; it is up to the author/donor to decide if he or she is willing to distribute software through any other system and a licensee cannot impose that choice.

This section is intended to make thoroughly clear what is believed to be a consequence of the rest of this License.

8. If the distribution and/or use of the Program is restricted in certain countries either by patents or by copyrighted interfaces, the original copyright holder who places the Program under this License may add an explicit geographical distribution limitation excluding those countries, so that distribution is permitted only in or among countries not thus excluded. In such case, this License incorporates the limitation as if written in the body of this License.

9. The Free Software Foundation may publish revised and/or new versions of the General Public License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns.

Each version is given a distinguishing version number. If the Program specifies a version number of this License which applies to it and "any later version", you have the option of following the terms and conditions either of that version or of any later version published by the Free Software Foundation. If the Program does not specify a version number of this License, you may choose any version ever published by the Free Software Foundation.

10. If you wish to incorporate parts of the Program into other free programs whose distribution conditions are different, write to the author to ask for permission. For software which is copyrighted by the Free Software Foundation, write to the Free Software Foundation; we sometimes make exceptions for this. Our decision will be guided by the two goals of preserving the free status of all derivatives of our free software and of promoting the sharing and reuse of software generally.

NO WARRANTY

11. BECAUSE THE PROGRAM IS LICENSED FREE OF CHARGE, THERE IS NO WARRANTY
FOR THE PROGRAM, TO THE EXTENT PERMITTED BY APPLICABLE LAW. EXCEPT WHEN
OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES
PROVIDE THE PROGRAM "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED
OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF
MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS
TO THE QUALITY AND PERFORMANCE OF THE PROGRAM IS WITH YOU. SHOULD THE
PROGRAM PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING,
REPAIR OR CORRECTION.

12. IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING
WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MAY MODIFY AND/OR
REDISTRIBUTE THE PROGRAM AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES,
INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING
OUT OF THE USE OR INABILITY TO USE THE PROGRAM (INCLUDING BUT NOT LIMITED
TO LOSS OF DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY
YOU OR THIRD PARTIES OR A FAILURE OF THE PROGRAM TO OPERATE WITH ANY OTHER
PROGRAMS), EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE
POSSIBILITY OF SUCH DAMAGES.

END OF TERMS AND CONDITIONS