**MSE ASSIGNMENT**

**Name: Jayavardhini S**

**Usn : 1PI10IS040**

**Class : 7 ‘A’ ISE**

**1)** **Explore the DVM instructions and prepare a summary of the same atleast for 5 instructions in a detailed format**

**i) instruction name:**

**ii) syntax**

**iii)example**

(1)

i) Instruction name: MOVE

ii)Syntax : move vx, vy //Moves the content of vy into vx. Both registers must be in the first 256 register range.

iii) Example: 0110 - move v0, v1  
 Moves v1 into v0.

(2)

i) Instruction name: return-object

ii) Syntax : return-object vx // Return with vx object reference value.

iii)Example : 1100 - return-object v0  
 Returns with object reference value in v0

(3)

i) Instruction name: Check the instance of class

ii) Syntax : instance-of vx,vy,type\_id // Checks whether vy is instance of a class identified by type\_id. Sets vx non-zero if it is 0 otherwise.

iii)Example: 2040 0100 - instance-of v0, v4, Test3 // type@0001  
Checks whether the object reference in v4 is an instance of type@0001 (entry #1 in the type id table). Sets v0 to non-zero if v4 is instance of Test3, 0 otherwise.

(4)

i) Instruction name: Array Length

ii) Syntax: array-length vx,vy // Calculates the number of elements of the array referenced by vy and puts the length value into vx.

iii) Example : 2111 - array-length v1, v1  
 Calculates the number of elements of the array referenced by v1 and puts the result into v1.

(5)

i) Instruction name: goto statement

ii) Syntax : goto target // Unconditional jump by short offset.

iii) Example : 28F0 - goto 0005 // -0010  
 Jumps to current position-16 words (hex 10). 0005 is the label of the target instruction.

**2) Differentiate between mobile and cloud computing?**

**Mobile computing** is the ability to use computing capability without a pre-defined location and/or connection to a network to publish and/or subscribe to information. **In simple terms** taking a physical device with you. This could be a laptop or a mobile phone or some device which enables you to telework – working wherever you go because of the small size of the device you’re using.

It is human–computer interaction by which a computer is expected to be transported during normal usage. Mobile computing involves mobile communication, mobile hardware, and mobile software. Communication issues include ad hoc and infrastructure networks as well as communication properties, protocols, data formats and concrete technologies. Hardware includes mobile devices or device components. Mobile software deals with the characteristics and requirements of mobile applications.

**Characteristic Requirements:**

* Mobile cloud computing needs to overcome mobile device differences.
* Mobile cloud computing needs to allow for disconnected operation
* Mobile cloud computing needs to be communication fault tolerant
* Distance matters in mobile cloud computing.
* Mobile cloud computing needs to be mindful of limited energy availability on mobiles.
* Expanded testing capabilities

**Advantages:**

* Increase in Productivity- Mobile devices can be used out in the field of various companies, therefore reducing the time and cost for clients and themselves.
* Entertainment- Mobile devices can be used for entertainment purposes, for personal and even for presentations to people and clients.
* Portability- this would be one of the main advantages of mobile computing, you are not restricted to one location in order for you to get jobs done or even access email on the go

**Limitations:**

* Range & Bandwidth
* Security standards
* Power consumption
* Transmission interferences
* Potential health hazards
* Human interface with device

**Conclusion:**

Mobile apps demand a lot more from the mobile cloud than regular cloud computing. Most of these differences are due to the limited energy availability, network latencies and unreliable connectivity in mobile devices. Fortunately, recognizing these differences and adding additional capabilities to the mobile cloud to address them, you can deliver as good an end user experience on mobiles, as you can on desktops and laptops.

**Cloud Computing:**

**Cloud computing** one of the new buzz words of 2008, allows you to store your files and folders in a “cloud” area on the Internet, allowing you access to all of your files and folders wherever you are in the world – but you do need a physical device with Internet access to access it.

Also one can have all files synchronised between devices so wherever one goes, can always have access to his/her files.

**Characteristic Requirements:**

* ‘On-Demand, Self-Service’ implies a customer can order service via the web or some other method at any point in time, 24×7, which becomes immediately available for his or her use.
* ‘Broad network access’ implies widespread, heterogeneous network accessibility for thin, thick, mobile and other commonly used compute mediums.
* ‘Shared resource pooling’ connotes the aggregation of physical compute resources into a logical ‘pool’ that is dynamically allocated in a multi-tenancy capacity across broad application service requirements.
* ‘Rapid, bi-directional elasticity’ simply means additional capacity remains available and accessible on an ‘as needed’ basis, and is recovered back to the pool when no longer needed for alternative allocation.
* ‘Metered service’ as noted above means that all variables of resource consumption are tracked in capacity that users can be automatically billed for their consumption.

**Advantages:**

* Cost Efficient : Cloud computing is probably the most cost efficient method to use, maintain and upgrade.
* Almost Unlimited Storage : Storing information in the cloud gives you almost unlimited storage capacity.
* Backup and Recovery : Since all your data is stored in the cloud, backing it up and restoring the same is relatively much easier than storing the same on a physical device.
* Automatic Software Integration : In the cloud, software integration is usually something that occurs automatically.
* Easy Access to Information : Once you register yourself in the cloud, you can access the information from anywhere, where there is an Internet connection.
* Quick Deployment : Lastly and most importantly, cloud computing gives you the advantage of quick deployment.

**Disadvantages :**

* Technical Issues : Though it is true that information and data on the cloud can be accessed anytime and from anywhere at all, there are times when this system can have some serious dysfunction.
* Security in the Cloud : The other major issue while in the cloud is that of security issues. A virus, for instance, hiding in the "cloud" would be able to infect every local computer that connects to it.
* Prone to Attack : Storing information in the cloud could make your company vulnerable to external attacks and threats.

**Conclusion :**

Like everything else, cloud computing too has its pros and cons. While the technology can prove to be a great asset to your company, it could also cause harm if not understood and used properly. Cloud computing relies on sharing of resources to achieve coherence and economies of scale, similar to a utility (like the electricity grid) over a network. At the foundation of cloud computing is the broader concept of converged infrastructure and shared services. The cloud also focuses on maximizing the effectiveness of the shared resources.

**3) Give an example of an application simulating an environment of context aware computing and justify.**

**Context aware app - The CybreMinder**

To aid our investigation of reminder tools and interfaces, we built the Java-based CybreMinder tool. It has two main parts — reminder creation and reminder delivery.

**1)Reminder Creation**

When users launch CybreMinder, they are presented with an interface that looks quite similar to an e-mail creation tool. Users can enter the names of the recipients for the reminder. The recipients could just be themselves, indicating a personal reminder, or a list of other people, indicating a third party reminder is being created. The reminder has a subject, a priority level (ranging from lowest to highest), a body in which the reminder description is placed, and an expiration date. The expiration date indicates the date and time at which the reminder should expire and be delivered, if it has not already been delivered.

**2)Reminder Delivery**

When a reminder can be delivered, either because its associated situation was satisfied or because it has expired, CybreMinder determines what is the most appropriate delivery mechanism for each reminder recipient. The default signal is to show the reminder on the closest available display, augmented with an audio cue. However, if a recipient wishes, they can specify a configuration file that will override this default.

Currently, we support the delivery of reminders via SMS on a mobile phone, e-mail, displaying on a nearby display (wearable, handheld, or static CRT) and printing to a local printer (to emulate paper to-do lists).

**Example Reminders:**

|  |  |  |
| --- | --- | --- |
| Situation  (Reminders) | Natural Language  Description | CybreMinder Description |
| Time | 9:45 am | Expiration field: 9:45 am |
| Location | Forecast is for rain and  Bob is leaving home | City = Atlanta, WeatherForecast = rain  Username = Bob, Location = Bob’s front door |
| Co-Location | Sally and colleague are  co-located | Username = Sally, Location = \*1  Username = Bob, Location = \*1 |
| Complex #1 | Stock price of X is over  $50, Bob is alone and  has free time | StockName = X, StockPrice > 50  Username = Bob, Location = \*1  Location = \*1, OccupantSize = 1  Username = Bob, FreeTime > 30 |
| Complex #2 | Sally is in her office has  some free time, and her  friend is not busy | Username = Sally, Location = Sally’s office  Username = Sally, FreeTime = 60  Username = Tom, ActivityLevel = low |

**Table 1.** Natural language and CybreMinder descriptions of scenarios discussed below

**Features:**

**Time-Based Reminder**

Like many of the other systems previously described, CybreMinder allows reminders to be triggered based on a simple time context. In this scenario, Sally has a meeting at 10 a.m. tomorrow. She wants to send a reminder to herself fifteen minutes before the meeting occurs, so that she has time to walk to the meeting. She can simply set the expiry date to be tomorrow’s date and 9:45 a.m.

**Location-Based Reminder**

In this scenario, Bob wants to remind himself to take his umbrella to work because itis supposed to rain this afternoon. He keeps the umbrella near his apartment door, sohe wants to receive the reminder as he approaches the door. Here, he can simply createa situation with only one sub-situation: he is at his front door. In CybreMinderterms, he sets the username to his name and location to his front door. This situationcan be made slightly more complex. If Bob is sending the reminder the night before,then he may want to add a time attribute and set it to be greater than 7:00 a.m. Bydoing so, the reminder will not be triggered and displayed each time he leaves his apartment that night. It will only be displayed when he approaches the door after 7:00a.m. the next morning. Pushing on this scenario a little more, Bob does have to knowahead of time that it is going to rain. He can simply create a reminder that is to bedelivered whenever the forecast calls for rain and he is leaving his apartment.

**Co-location-Based Reminder**

Of the systems we reviewed, only Proem [11] supported proactive reminders when two or more people were co-located in an arbitrary location. It can be argued that post-it notes could be used in this setting, although it currently breaks normal social conventions to stick post-it notes to people. An example co-location scenario follows: Sally wants to engage a colleague in a discussion about an interesting paper she read, but forgets when she sees her colleague. She can create a context-aware reminder that will be delivered when she is in close proximity with her colleague. The situation she creates is slightly more complex than the ones we have discussed so far, and it makes use of variables. Variables allow users to create relationships between sub-situations. First Sally creates an initial sub-situation where she sets the user name to be her colleague’s name and the location to be variable (indicated in Table 1 by \*1). Then, she creates a second sub-situation, where she sets the user name to be her name and the location to the variable used in the first sub-situation. Now when Sally and her colleague are in the same arbitrary location, the reminder will be delivered.

**Complex Reminder**

CybreMinder supports the unlimited use of rich context, allowing users to create as rich a situation as can be sensed. We describe two such situations. In the first scenario, Bob owns stock in Company X and has decided to sell that stock when it is valued over $50 per share. He only wants to be reminded to sell, however, when he is alone and has free time. To create this situation to signal a reminder to sell, Bob creates a number of sub-situations: stock price of company X > $50, Bob is the only occupant of his location, and Bob’s schedule shows that he has > 30 minutes before his next meeting. When this situation occurs, Bob receives the reminder to sell his stock. In our second complex scenario, Sally needs to make a phone call to her friend Tom. She wants to receive a context-aware reminder when she arrives at her office, has some free time in her schedule, and her friend is not busy. To create this situation, she creates three sub-situations: Sally is in her office, Tom’s activity status is low, and Sally has at least one hour before her next appointment.

**Concluding Justification on CybreMinder - simulating an environment of context aware computing:**

As seen in the example above, it behaves based on current context i.e., user sets a reminder based on different context e.g.: time, temperature, location etc, and when current context becomes equal to the saved context the reminder pops with the saved description. It keeps sensing based on context, i.e, in case the context is temperature, it keeps getting temperature information, and if context is location, it uses GPS to find latitude and longitude and spot the location using Map services. Hence justifying CybreMinder has a simulating environment of context aware computing.