**MSE-ASSIGNMENT**

**\_\_\_ JAMUNA G**

**1PI10IS039 Q1> 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**

Ans:

Instructions explained below are grouped according to the type of operations they perform:

* **Move instruction**

1. Instruction name: move
2. Syntax:
3. **move vA, vB** - Move the contents of one non-object register to another.

A: destination register (4 bits) B: source register (4 bits)

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

1. **move/from16 vAA, vBBBB** - Move the contents of one non-object register to another. A: destination register (8 bits)

B: source register (16 bits)

(iii)Example: 0200 1900 - move/from16 v0, v25  
 Moves v25 into v0.

1. **move-result vAA -** Move the result value of the previous method invocation into vx.

**(iii)Example:** 0A00 - move-result v0  
 Move the return value of a previous method invocation into v0.

1. **move-object vA, vB** - Move the contents of one object-bearing register to another. A: destination register (4 bits) B: source register (4 bits)

(iii)Example: 0781 - move-object v1, v8  
 Moves the object reference in v8 to v1.

* **Stack operations**: Used to push or pop arguments from the stack.

1. Instruction name: PUSH is used to push 8-bit integer arguments (0 <= x <= 255) only. Thus, PUSH is a two byte instruction (1byte opcode + 1byte argument).
2. Syntax: **PUSH x** // wher x is a value.
3. Example: PUSH 10 // Push 10 onto the stack.

* **Math operations**:

**(1)**

(i)Instruction Name: Add

(ii)Syntax:

1) **add-int vx,vy,vz** //Adds vy to vz and puts the result into vx.

(iii)Example: 9000 0203 - add-int v0, v2, v3  
 Adds v3 to v2 and puts the result into v0.

2) **add-long vx, vy, vz** //Adds vy to vz and puts the result into vx.

(iii)Example: 9B00 0305 - add-long v0, v3, v5  
 The long value in v3,v4 is added to the value in v5,v6 and the result is stored in v0,v1.

3) **add-float vx,vy,vz** //Adds vy to vz and puts the result into vx.

(iii)Example: A600 0203 - add-float v0, v2, v3  
 Adds the floating point numbers in v2 and v3 and puts the result into v0.

(2)

(i)Instruction Name: Sub

(ii)Syntax:

1) **sub-int vx,vy,vz** //Calculates vy-vz and puts the result into vx.

(iii)Example: 9100 0203 - sub-int v0, v2, v3  
 Subtracts v3 from v2 and puts the result into v0.

2) **sub-long vx,vy,vz** //Calculates vy-vz and puts the result into vx.

(iii)Example: 9C00 0305 - sub-long v0, v3, v5  
 Subtracts the long value in v5,v6 from the long value in v3,v4 and puts the result into v0,v1.

3) **sub-float vx,vy,vz** //Calculates vy-vz and puts the result into vx.

(iii)Example: A700 0203 - sub-float v0, v2, v3  
 Calculates v2-v3 and puts the result into v0.

* **Comparison**
  + 1. Instruction name: compare
    2. Syntax:
       1. **cmpl-float** //Compares the float values in vy and vz and sets the integer value in vx accordingly

(iii)Example: cmpl-float v0, v6, v7  
//Compares the float values in v6 and v7 then sets v0 accordingly. NaN bias is less-than, the instruction will return -1 if any of the parameters is NaN.

* + - 1. **cmpg-float vx, vy, vz** //Compares the float values in vy and vz and sets the integer value in vx accordingly

(iii)Example: cmpg-float v0, v6, v7  
//Compares the float values in v6 and v7 then sets v0 accordingly. NaN bias is greater-than, the instruction will return 1 if any of the parameters is NaN.

* + - 1. **cmpl-double vx,vy,vz** //Compares the double values in vy and vz and sets the integer value in vx accordingly

(iii)Example: cmpl-double v25, v6, v8  
//Compares the double values in v6,v7 and v8,v9 and sets v25 accordingly. NaN bias is less-than, the instruction will return -1 if any of the parameters is NaN.

* + - 1. **cmpg-double vx, vy, vz** //Compares the double values in vy and vz and sets the integer value in vx accordingly

(iii)Example: cmpg-double v0, v8, v10  
//Compares the double values in v8,v9 and v10,v11 then sets v0 accordingly. NaN bias is greater-than, the instruction will return 1 if any of the parameters is NaN.

* + - 1. **cmp-long vx, vy, vz** //Compares the long values in vy and vz and sets the integer value in vx accordingly

(iii)Example: cmp-long v0, v2, v4  
//Compares the long values in v2 and v4 then sets v0 accordingly.

* **GOTO**

1. Instruction name: goto
2. Syntax:
3. **goto target** // Unconditional jump by short offset

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

1. **goto/16 target** // Unconditional jump by 16 bit offset

(iii)Example: goto/16 002f // -01f1  
 Jumps to the current position-1F1H words. 002F is the label of the target instruction.

**Q2> Differentiate between mobile and cloud computing**

Ans:

|  |  |  |
| --- | --- | --- |
|  | **MOBILE COMPUTING** | **CLOUD COMPUTING** |
| **DEFINITION** | 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.  Mobile computing involves [mobile communication](http://en.wikipedia.org/wiki/Mobile_communication), mobile hardware, and mobile software. | Cloud computing is a type of computing that relies on sharing computing resources rather than having local servers or personal [devices](http://www.webopedia.com/TERM/D/device.html) to handle [applications](http://www.webopedia.com/TERM/A/application.html).  cloud computing is a synonym for [distributed computing](http://en.wikipedia.org/wiki/Distributed_computing) over a network, |
| **CHARACTERISTICS** | PortabilitySocial InteractivityContext SensitivityConnectivity**Individuality** | * Agility * [Device and location independence](http://en.wikipedia.org/wiki/Device_independence) * [Virtualization](http://en.wikipedia.org/wiki/Virtualization) * [Reliability](http://en.wikipedia.org/wiki/Reliability_(computer_networking)) * [Performance](http://en.wikipedia.org/wiki/Computer_performance) * [Security](http://en.wikipedia.org/wiki/Computer_security) |
| **CHALLENGES/LIMITATIONS/ISSUES** | * Range & Bandwidth * Power consumption * Security standards * Transmission interferences * Potential health hazards * Human interface with device | Threats and opportunities of the cloudPrivacyOpen sourceSustainabilityPerformance interference and noisy neighbours |
| **APPLICATIONS** | * Emergency Services * Stock information collation/control * Credit card verification * Taxi/Truck dispatch * Electronic mail/paging * Web Access * Entertainment | GmailGoogle CalendarGoogle GroupsGoogle DocsEvernoteCampfireDropbox |
| **ADVANTAGES** | * Location flexibility * Saves Time * Enhanced Productivity * Ease of research * Streamlining of Business Processes | * Cost Efficient * Almost Unlimited Storage * Backup and Recovery * Automatic Software Integration * Flexibility * Quick Deployment |

**Q3> Give an example of an application simulating an environment of context aware computing and justify.**

Ans :

Context-aware computing deals with the ability of computer systems to take advantage of information from or conditions in the dynamic environment to provide added-value services or to execute more complex tasks.

Example of context aware computing application:

**SenSay: A Context-Aware Mobile Phone**

* SenSay is a context-aware mobile phone that adapts to dynamically changing environmental and physiological states and also provides the remote caller information on the current context of the phone user.
* SenSay (sensing & saying) is a context-aware mobile phone that modifies its behavior based on its user's state and surroundings.
* To provide context information SenSay uses light, motion, and microphone sensors.
* In addition to manipulating ringer volume, vibration, and phone alerts, SenSay can provide remote callers with the ability to communicate the urgency of their calls, make call suggestions to users when they are idle, and provide the caller with feedback on the current status of the SenSay user.
* A number of sensors including accelerometers, light, and microphones are mounted at various points on the body to provide data about the user’s context. A decision module uses a set of rules to analyze the sensor data and manage a state machine composed of uninterruptible, idle, active and normal states.
* Results from our threshold analyses show a clear delineation can be made among several user states by examining sensor data trends.
* SenSay augments its contextual knowledge by tapping into applications such as electronic calendars, address books, and task lists.

## SatNav as Context-Aware system

* In a Satellite Navigation System (SatNav), the current location is the primary contextual parameter that is used to automatically adjust the visualization (e.g. map, arrows, directions…) to the user’s current location.
* However, looking at current commercial systems, much more context information is used and much of visualization has been changed.
* In addition to the current GPS position, contextual parameters may include the time of day, light conditions, the traffic situation on the calculated route or the user’s preferred places. Beyond the visualization and whether or not to switch on the backlight, the calculated route can be influenced by context, e.g. to avoid potentially busy streets at that time of day.