List data structure in FreeRTOS

Slides prepared by Sumesh Divakaran

Indian Institute of Science, Bangalore.

23 August 2011
The macros, function prototypes and structures are defined in `list.h`.

There are three important data structures namely `xList`, `xListItem` and `xMiniListItem` in list.
**xList**

- **xList** contains the header information regarding a list. Its structure is:

<table>
<thead>
<tr>
<th>uxNumberOfItems</th>
<th>pxIndex</th>
<th>xListEnd</th>
</tr>
</thead>
</table>

  **xList – Structure**

- **uxNumberOfItems** is the number of items present in the list.
- **pxIndex** is a pointer to an item (**xListItem**) in the list, which is used to traverse the list.
- **xListEnd** is a (**xMiniListItem**) which is used as an end-marker in the list.
**xListItem**

- **xListItem** is an node in an xList. This structure contains 5 elements:

<table>
<thead>
<tr>
<th>pxPrevious</th>
<th>pvOwner</th>
<th>xItemValue</th>
<th>pxContainer</th>
<th>pxNext</th>
</tr>
</thead>
</table>

**pxPrevious** is the pointer to the previous element (**xListItem**) in the list.

**pvOwner** is a pointer to the owner of the *item* in the list. E.g. it points to the TCB of a task if the list is the ready queue.

**xItemValue** is the value of the *item* in the list.

**pvContainer** is a pointer to the list (**xList**) in which the *item* is present.

**pxNext** is a pointer to the next element (**xListItem**) in the list.
**xMiniListItem**

- **xMiniListItem** is an element in a list used to mark the end of the list. The *xListEnd* in **xList** contains this element.

<table>
<thead>
<tr>
<th>pxPrevious</th>
<th>xItemValue</th>
<th>pxNext</th>
</tr>
</thead>
</table>

- **pxPrevious** is the pointer to the previous element (**xListItem**) in the list.
- **xItemValue** is the value of the *item* in the list.
- **pxNext** is a pointer to the next element (**xListItem**) in the list.
vListInitialse() is a function to initialise a list (*xList*).

Initial list (say ReadyQ):

<table>
<thead>
<tr>
<th><em>uxNumberOfItems</em></th>
<th><em>pxIndex</em></th>
<th><em>xListEnd</em></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

After `vListInitialse(readyQ)` ⇒:

```
0                      100
```

```
vListInitialiseItem()

- \textbf{vListInitialiseItem(xListItem *):} This function is used to initialise a list item (xListItem).

\begin{table}[h]
\centering
\begin{tabular}{|c|c|c|c|c|}
\hline
pxPrevious & pvOwner  & xItemValue & pxContainer & pxNext \\
\hline
\end{tabular}
\end{table}

\textbf{xListItem}

After \textbf{vListInitialiseItem(xItem)} \Rightarrow:

\begin{table}[h]
\centering
\begin{tabular}{|c|c|c|c|}
\hline
 & & & NULL \\
\hline
\end{tabular}
\end{table}
vListInsert()

- \textit{vListInsert(xList \ast, xListItem \ast)} inserts an item into a list in increasing order of item values.

After \textit{vListInsert(readyQ,xItem1)} :
vListInsert()
vListInsert()
vListInsert()

vListInsert(readyQ, xItem4) ⇒

1 2 2
\textbf{vListRemove()}

- \textit{vListRemove(xListItem *)} is a function to delete an item from the list.

\begin{itemize}
  \item \textbf{vListRemove(xItem3)} \Rightarrow
  \begin{itemize}
    \item Task1
    \item Task2
    \item Task3
    \item Task4
  \end{itemize}
\end{itemize}
vListRemove() is a function to delete an item from the list.

vListRemove(xListItem *)

Task1 Task2 Task3 Task4

readyQ:

vListRemove(xItem3) ⇒

Task1 Task2 Task3 Task4
\textbf{vListRemove()}

- \textit{vListRemove(xListItem * )} is a function to delete an item from the list.

\begin{itemize}
  \item \texttt{vListRemove(xItem3)} \Rightarrow
\end{itemize}

\begin{itemize}
  \item \texttt{vListRemove(xItem3)} \Rightarrow
\end{itemize}
vListRemove() is a function to delete an item from the list.

vListRemove(xListItem *) is a function to delete an item from the list.

vListRemove(xItem3) ⇒
vListInsertEnd()

- `vListInsertEnd(xList*, xListItem*)` inserts an item at the end of list. The new element becomes the last element to be returned by a call to `listGetOwnerOfNextEntry`.

```
someList: 3 [4] 100
          ├── 10 ─── 5 ─── 20 ─── 100

Task1    Task2    Task3
```

```
vListInsertEnd(someList, xItem4) ⇒
```

```
someList: 4 [100]
          ├── 10 ─── 5 ─── 15 ─── 20 ─── 100

Task1    Task2    Task4    Task3
```
list_GET_OWNER_OF_NEXT_ENTRY()

- `list_GET_OWNER_OF_NEXT_ENTRY(taskTCB *, xList *)` return the owner of the list item after the one pointed to by `pxIndex` in `xList`. `pxIndex` is also updated to point to next item in the list.
list_GET_OWNER_OF_NEXT_ENTRY()}

- `list_GET_OWNER_OF_NEXT_ENTRY(taskTCB *, xList *)` return the owner of the list item after the one pointed to by `pxIndex` in `xList`. `pxIndex` is also updated to point to next item in the list.

```
readyQ: 4 100
```

```
Task1 Task2 Task3 Task4
```

```
list_GET_OWNER_OF_NEXT_ENTRY(pxTCB, readyQ) ⇒
```
list_GET.OwnerOf.Next.Entry()

- list_GET.OwnerOf.Next.Entry(taskTCB *, xList *) return the owner of the list item after the one pointed to by pxIndex in xList. pxIndex is also updated to point to next item in the list.

\[
\text{list_GET.OwnerOf.Next.Entry(pxTCB, readyQ)} \implies \text{returns Task1 in pxTCB}
\]