



1997, 1998, 1999, 2000, 2001, 2002, 2003, 2004, 2005, 2006, 2007, 2008, 2009, 2010, 2011, 2012, 2013, 2014, 2015, 2016, 2017, 2018, 2019, 2020, 2021, 2022, 2023, 2024, 2025, 2026, 2027, 2028, 2029, 2030, 2031, 2032, 2033, 2034, 2035, 2036, 2037, 2038, 2039, 2040, 2041, 2042, 2043, 2044, 2045, 2046, 2047, 2048, 2049, 2050, 2051, 2052, 2053, 2054, 2055, 2056, 2057, 2058, 2059, 2060, 2061, 2062, 2063, 2064, 2065, 2066, 2067, 2068, 2069, 2070, 2071, 2072, 2073, 2074, 2075, 2076, 2077, 2078, 2079, 2080, 2081, 2082, 2083, 2084, 2085, 2086, 2087, 2088, 2089, 2090, 2091, 2092, 2093, 2094, 2095, 2096, 2097, 2098, 2099, 2100, 2101, 2102, 2103, 2104, 2105, 2106, 2107, 2108, 2109, 2110, 2111, 2112, 2113, 2114, 2115, 2116, 2117, 2118, 2119, 2120, 2121, 2122, 2123, 2124, 2125, 2126, 2127, 2128, 2129, 2130, 2131, 2132, 2133, 2134, 2135, 2136, 2137, 2138, 2139, 2140, 2141, 2142, 2143, 2144, 2145, 2146, 2147, 2148, 2149, 2150, 2151, 2152, 2153, 2154, 2155, 2156, 2157, 2158, 2159, 2160, 2161, 2162, 2163, 2164, 2165, 2166, 2167, 2168, 2169, 2170, 2171, 2172, 2173, 2174, 2175, 2176, 2177, 2178, 2179, 2180, 2181, 2182, 2183, 2184, 2185, 2186, 2187, 2188, 2189, 2190, 2191, 2192, 2193, 2194, 2195, 2196, 2197, 2198, 2199, 2200, 2201, 2202, 2203, 2204, 2205, 2206, 2207, 2208, 2209, 2210, 2211, 2212, 2213, 2214, 2215, 2216, 2217, 2218, 2219, 2220, 2221, 2222, 2223, 2224, 2225, 2226, 2227, 2228, 2229, 2230, 2231, 2232, 2233, 2234, 2235, 2236, 2237, 2238, 2239, 2240, 2241, 2242, 2243, 2244, 2245, 2246, 2247, 2248, 2249, 2250, 2251, 2252, 2253, 2254, 2255, 2256, 2257, 2258, 2259, 2260, 2261, 2262, 2263, 2264, 2265, 2266, 2267, 2268, 2269, 2270, 2271, 2272, 2273, 2274, 2275, 2276, 2277, 2278, 2279, 2280, 2281, 2282, 2283, 2284, 2285, 2286, 2287, 2288, 2289, 2290, 2291, 2292, 2293, 2294, 2295, 2296, 2297, 2298, 2299, 2300, 2301, 2302, 2303, 2304, 2305, 2306, 2307, 2308, 2309, 2310, 2311, 2312, 2313, 2314, 2315, 2316, 2317, 2318, 2319, 2320, 2321, 2322, 2323, 2324, 2325, 2326, 2327, 2328, 2329, 2330, 2331, 2332, 2333, 2334, 2335, 2336, 2337, 2338, 2339, 2340, 2341, 2342, 2343, 2344, 2345, 2346, 2347, 2348, 2349, 2350, 2351, 2352, 2353, 2354, 2355, 2356, 2357, 2358, 2359, 2360, 2361, 2362, 2363, 2364, 2365, 2366, 2367, 2368, 2369, 2370, 2371, 2372, 2373, 2374, 2375, 2376, 2377, 2378, 2379, 2380, 2381, 2382, 2383, 2384, 2385, 2386, 2387, 2388, 2389, 2390, 2391, 2392, 2393, 2394, 2395, 2396, 2397, 2398, 2399, 2400, 2401, 2402, 2403, 2404, 2405, 2406, 2407, 2408, 2409, 2410, 2411, 2412, 2413, 2414, 2415, 2416, 2417, 2418, 2419, 2420, 2421, 2422, 2423, 2424, 2425, 2426, 2427, 2428, 2429, 2430, 2431, 2432, 2433, 2434, 2435, 2436, 2437, 2438, 2439, 2440, 2441, 2442, 2443, 2444, 2445, 2446, 2447, 2448, 2449, 2450, 2451, 2452, 2453, 2454, 2455, 2456, 2457, 2458, 2459, 2460, 2461, 2462, 2463, 2464, 2465, 2466, 2467, 2468, 2469, 2470, 2471, 2472, 2473, 2474, 2475, 2476, 2477, 2478, 2479, 2480, 2481, 2482, 2483, 2484, 2485, 2486, 2487, 2488, 2489, 2490, 2491, 2492, 2493, 2494, 2495, 2496, 2497, 2498, 2499, 2500, 2501, 2502, 2503, 2504, 2505, 2506, 2507, 2508, 2509, 2510, 2511, 2512, 2513, 2514, 2515, 2516, 2517, 2518, 2519, 2520, 2521, 2522, 2523, 2524, 2525, 2526, 2527, 2528, 2529, 2530, 2531, 2532, 2533, 2534, 2535, 2536, 2537, 2538, 2539, 2540, 2541, 2542, 2543, 2544, 2545, 2546, 2547, 2548, 2549, 2550, 2551, 2552, 2553, 2554, 2555, 2556, 2557, 2558, 2559, 2560, 2561, 2562, 2563, 2564, 2565, 2566, 2567, 2568, 2569, 2570, 2571, 2572, 2573, 2574, 2575, 2576, 2577, 2578, 2579, 2580, 2581, 2582, 2583, 2584, 2585, 2586, 2587, 2588, 2589, 2590, 2591, 2592, 2593, 2594, 2595, 2596, 2597, 2598, 2599, 2600, 2601, 2602, 2603, 2604, 2605, 2606, 2607, 2608, 2609, 2610, 2611, 2612, 2613, 2614, 2615, 2616, 2617, 2618, 2619, 2620, 2621, 2622, 2623, 2624, 2625, 2626, 2627, 2628, 2629, 2630, 2631, 2632, 2633, 2634, 2635, 2636, 2637, 2638, 2639, 2640, 2641, 2642, 2643, 2644, 2645, 2646, 2647, 2648, 2649, 2650, 2651, 2652, 2653, 2654, 2655, 2656, 2657, 2658, 2659, 2660, 2661, 2662, 2663, 2664, 2665, 2666, 2667, 2668, 2669, 2670, 2671, 2672, 2673, 2674, 2675, 2676, 2677, 2678, 26

# Magnetic Strip / Card Readers (MSR / MCR)



- § Triple track card readers
- § Simultaneous read of all tracks
- § Swipe up or down
- § Decoding based upon ISO 7811, AAMVA, and California DL/ID specifications

# Magnetic Stripe Card Coding



## § Track 1     **ISO 3554**

- IATA     (International Air Trans. Assn.)
- 79 Alphanumeric chars., read only

## § Track 2     **ISO 3554**

- ABA (American Banking Assn.)
- 40 Numeric chars., read only

## § Track 3     **ISO 4909**

- ATM or Thrift Industry (ISO 4909)
- 107 Numeric chars., read/write

# Device Usage in Applications



```
int hCard, bytes_read, bytes_written,  
    status, iResult;  
open  
    hCard = open(DEV_CARD, 0);  
close  
    status = close(hCard);  
read  
    bytes_read = read(hCard, buffer, size);  
write  
    bytes_written = write(hCard, buffer, size);
```

# Data from read( )



- § Only one card scan can be queued at a time.
- § Data may not exist for all tracks
- § Card buffer should be large enough to hold all data
- § Always receive Count ( $C_x$ ) and Status ( $S_x$ ) bytes for all three tracks.
- § The status byte indicates the status of the track data.
- § Format of data in application buffer is as follows:  
$$C_1 S_1 D_1 C_2 S_2 D_2 C_3 S_3 D_3$$
- § Each track is decoded

# Reading Raw Data



- § `int card_raw_data(char *buffer)`
- § Returns the raw contents of the card swipe buffer before decoding
- § Does not clear the card swipe buffer
- § The card swipe is still pending
- § Buffer should be at least 290 bytes

# Pending Card Data?



§ Two functions available for determining if card data is available to be read from card reader

- `int card_pending(void)`
- `int chk_card_rdr(int hCard)`

§ Getting account information from keyboard or card reader

- `key_card_entry( )`

# Parsing Card Data



```
§ return = card_parse(datain, &trackdata,  
    "2");
```

Parses data from an application buffer to a TRACK structure

§ Pass the data from the card read

§ Pointer to a structure

§ Track selection for parsing (track 2 in this example)



# Related Card Reader Functions



```
int  chk_luhn(acct_num) ;
```

Accepts a NULL terminated string containing digits, and verifies that the last digit is the correct MOD 10 check digit.

```
int  SVC_INFO_MAG(char *buff)
```

Stores a 1-byte magnetic card reader type code in the caller's buffer.

# LAB Exercise



## LAB7: Magnetic Card Reader

Make use of the magnetic stripe reader device to obtain information about the user from the user's magnetic stripe card.

.

# Clock Device



- § Maintains the current date and time
- § Provides a source of periodic interrupts for system timing
- § Date and time format:
  - Read 15 bytes. `yyyymmddhhmmssd`
  - Write 14 bytes. `yyyymmddhhmmss`
- § Clock does not need to be open to read
  - Use `read_clock()`
- § Only one application can open the clock
  - `hClock = open(DEV_CLOCK, 0)`
- § Once the clock has been opened it may be read from or written to
  - `read()`
  - `write()`

# Date/Time Functions



## § Convert a date time string to integer values

- `int clock2int(hClock, buffer, &yr, &mth, &day, &hour, &min, &sec, &wday)`

## § Convert a specified date to a Julian value

- `int julian_date(year, month, day)`

## § Determine if the year specified is a leap year

- `int LEAP_YR(year)`

## § Returns the number of days in the month for the specified year

- `int month_end(month, year)`

## § Provide read/write access to the system clock

- `int clock_data(clock, I_type, clock_buffer)`

# Application Wait Functions



§ `void SVC_WAIT(time);`

Causes a non-interruptible software delay.

Time: 10-65535 ms

§ `void SLEEP(time);`

Causes a non-interruptible software delay.

Time: 10-ms increments. One minute maximum.

§ `void pause(time);`

Causes a software delay, interruptible by keypress.

Time: 10-ms increments

# Using Timeouts



- § Get the current clock value
- § Calculate stop time
- § Wait for timer to expire

# Timeout Functions



§ `int SVC_TICKS(action, long start)`

- Read the system's tick counter

§ `SVC_CLOCK();`

§ `int tick_compare(cur_tick, end_tick)`

- Compare two timer values

§ `unsigned long set_itimeout(hClock, time, interval)`

- Set an interval timer

§ `int CHK_TIMEOUT(hClock, time_val)`

- Compares a timer value to the current tick count

# Internal Thermal Printer



- § The ITP is a thermal array printer that communicates with the host terminal through the RS-232 port. The data format of the ITP is fixed at 8 bits, no parity, stop bit (8N1) at 19200 bps.
- § The com port assigned to ITP is fixed as COM4



# Internal Thermal Printer



§ The ITP has the following additional features:

- Downloadable user fonts
- Downloadable user logo

| Font  | Size | Bytes per character | Font table required |
|-------|------|---------------------|---------------------|
| 5X8   | 2    | 8                   | 1                   |
| 8X14  | 3    | 14                  | 2                   |
| 16X16 | 1    | 32                  | 4                   |
| 24X24 | 5    | 72                  | 9                   |

It has 64 font tables used to download user-defined fonts for printing.

# Printer Functions



- § p3300\_init() --Initializes the printer
- § p3300\_dnld\_font\_file() --Downloads a font file set to the printer. The font file is composed of a 8-byte header record.
- § p3300\_dnld\_graphic\_file() -- Downloads a graphic logo file into the printer. The graphic file is composed of a 16-byte header record.

# Printer Functions



- § p3300\_select\_font() --Selects the font table to use for printing or downloading.
- § p3300\_status() --Queries the printer for its status.
- § p3300\_print() --Sends a text string to the printer
- § p3300\_close() --Close printer.

# Printer attribute



**PRINT\_NORM** (Normal)

**DEL\_HEIGHT** (Double hight)

**DBL\_WIDTH** (Double width)

**SEL\_HEX** (hex format location)

**INVERSE** (inverse )

# LAB Exercise



## **LAB8:** Real time Clock and Printer

Read the clock to determine the current date and use it in transaction processing. Print the transaction data in a raw form. (print out company's logo)