Given a string $S$, the compressed indexing problem is to preprocess $S$ into a compressed representation that supports fast substring queries. The goal is to use little space relative to the compressed size of $S$ while supporting fast queries. We present a compressed index based on the Lempel-Ziv 1977 compression scheme. Let $n$, and $z$ denote the size of the input string, and the compressed LZ77 string, respectively. We obtain the following time-space trade-offs. Given a pattern string $P$ of length $m$, we can solve the problem in (i) $O(m + \text{occ} \cdot \lg \lg n)$ time using $O(z \cdot \lg(n/z) \cdot \lg \lg z)$ space, or (ii) $(m (1 + \lg \frac{z}{\lg(n/z)} + \text{occ} \cdot \lg \frac{n}{z}))$ time using $O(z \cdot \lg(n/z))$ space, for any $0 < \delta < 1$. In particular, (i) improves the leading term in the query time of the previous best solution from $O(m \cdot \lg m)$ to $O(m)$ at the cost of increasing the space by a factor $\lg \lg z$. Alternatively, (ii) matches the previous best space bound, but has a leading term in the query time of $O(m(1 + \lg \frac{z}{\lg(n/z)}))$. However, for any polynomial compression ratio, i.e., $z = O(n^{1-\delta})$, for constant $\delta > 0$, this becomes $O(m)$. Our index also supports extraction of any substring of length $\ell$ in $O(\ell + \lg(n/z))$ time. Technically, our results are obtained by novel extensions and combinations of existing data structures of independent interest, including a new batched variant of weak prefix search.